

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

FACULTY OF MECHANICAL ENGINEERING

INSTITUTE OF SOLID MECHANICS, MECHATRONICS

AND BIOMECHANICS

# UNIVERZÁLNÍ ŘÍDÍCÍ JEDNOTKA TECHNOLOGICKÝCH PROCESŮ

UNIVERSAL CONTROL UNIT OF TECHNOLOGICAL PROCESSES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MAREK GOGOLA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PETR KREJČÍ, Ph.D.

## **Abstrakt**

Táto diplomová práca sa zaoberá návrhom a realizáciou univerzálnej riadiacej jednotky technologických procesov v potravinárskom priemysle.

Úlohou tejto práce je vytvoriť zariadenie, ktoré je schopné vzhľadom na žiadanú veličinu a pomocou nameraných dát samostatne regulovať výstup. V tejto práci bola vykonaná realizácia riadiacej jednotky ako aj z hardvérového, tak aj zo softvérového hľadiska. Riadiaca jednotka je schopná samostatne meniť výstupný výkon podľa nameranej teploty. Ďalej slúži na ovládanie dvoch relé. Na zadanie žiadaných veličín a ovládanie relé sa používa riadiaci softvér v PC.

## **Abstract**

This diploma thesis is about design and realization of universal control unit of technologic processes in food-processing industry. The main goal of this work is to create a device, which will be able with respect to desired values and using measured data autonomously regulate output. In this work was done the realization of the control unit both from hardware and software aspect. Designed control unit is able to change the output power according to measured temperature. Furthermore is able to control of 2 relays. To set up the desired values and control of relays is used PC software.





Za podporu, neoceniteľné rady a trpezlivosť ďakujem vedúcemu diplomovej práce Ing. Petrovi Krejčímu, Ph.D. a ostatným ľuďom, ktorí sa o vznik tejto práce pričínili.

Bc. Marek Gogola



Prehlasujem, že túto diplomovú prácu som vypracoval samostatne pod vedením vedúceho diplomovej práce a s použitím uvedenej literatúry.

Marek Gogola, Brno, 2010

Bc. Marek Gogola

# Obsah

<b>1</b>	<b>Úvod</b>	<b>11</b>
<b>2</b>	<b>Formulácia problému a ciele riešenia</b>	<b>12</b>
<b>3</b>	<b>Rešeršné štúdium</b>	<b>15</b>
3.1	Riadiace jednotky . . . . .	15
<b>4</b>	<b>Analýza problému</b>	<b>21</b>
4.1	Riadiaca časť . . . . .	21
4.1.1	Mikročip . . . . .	21
4.1.2	Regulácia, typy regulácie a regulátorov . . . . .	23
4.1.3	Možnosti regulácie odporovej záťaže . . . . .	28
4.1.4	Snímače teploty . . . . .	30
<b>5</b>	<b>Realizácia riadiacej jednotky</b>	<b>34</b>
5.1	Riadiaca časť . . . . .	34
5.1.1	Mikrokontrolér Atmel AVR Atmega 8 . . . . .	34
5.1.2	LCD displej . . . . .	38
5.1.3	Sériová komunikácia . . . . .	38
5.1.4	Čidlo DS18B20 . . . . .	42
5.2	Výkonová časť . . . . .	48
<b>6</b>	<b>Prezentácia výsledkov</b>	<b>50</b>
6.1	Realizácia plošného spoja . . . . .	50
6.1.1	Výkonová časť . . . . .	50
6.1.2	Riadiaca časť . . . . .	51
6.2	Namerané hodnoty . . . . .	52
<b>7</b>	<b>Záver</b>	<b>55</b>
<b>8</b>	<b>Literatúra a odkazy</b>	<b>56</b>
<b>9</b>	<b>Zoznam príloh</b>	<b>60</b>



<b>10 Prílohy</b>	<b>61</b>
10.1 Príloha č.1 - schéma výkonovej časti dosky . . . . .	61
10.2 Príloha č.2-doska plošných spojov výkonovej časti . . . . .	62
10.3 Príloha č.3 - schéma riadiacej časti dosky . . . . .	63
10.4 Príloha č.4 - doska plošných spojov riadiacej časti . . . . .	64
10.5 Príloha č.5 - výpis zdrojového kódu teplotného čidla DS18B20 . . . . .	65
10.6 Príloha č.6 - výpis zdrojového kódu pre obsluhu USART . . . . .	68
10.7 Príloha č.7 - výpis zdrojového kódu pre obsluhu LCD . . . . .	72

## Zoznam použitých veličín

P	Výkon ohrievacieho telesa	[W]
$\alpha$	Riadiaci uhol	[°]
U	Sieťové napätie	[V]
$U_{cc}$	Napájacie napätie riadiacej časti	[V]
$U_{in}$	Vstupné napätie stabilizátora	[V]
$U_{out}$	Výstupné napätie stabilizátora	[V]
I	Zaťažovací prúd	[A]
f	Frekvencia sieťového napätia	[Hz]
$f_{osc}$	Frekvencia mikrokontroléra	[Hz]
T	Meraná teplota	[°C]
t	Meraný čas	[s]
y	Regulovaná veličina	[—]
w	Žiadaná veličina	[—]
e	Regulačná odchýlka	[—]
u	Akčná veličina	[—]
$K_P$	Proporcionálna konštanta regulátora	[—]
$K_I$	Integračná konštanta regulátora	[—]
$K_D$	Derivačná konštanta regulátora	[—]
bit	Binary digit- dvojková sústava s hodnotou 0, alebo 1	[—]
byte	Bajt, jednotka kapacity, slovo s dĺžkou 8 bitov	[—]
LSB	Least Significant Byte, najnižší byte slova	[—]
MSB	Most Significant Byte, najvyšší byte slova	[—]
RISC	Reduced Instruction Set Computer - počítač s redukovanou inštrukčnou sadou	[—]

# 1. Úvod

V priebehu vývoja spoločnosti sa človek najprv podľa svojich schopností, možností a záujmu začal oslobodzovať od namáhanej a opakujúcej sa fyzickej práce (napr. prechodom z ručného na strojné obrábanie). Neskôr, s ďalším rozvojom techniky a nárastom nárokov na riadiacu činnosť, pristúpil i k oslobodeniu od často už i veľmi náročnej a namáhavej riadiacej duševnej práce (napr. prechod od strojného obrábania s ľudskou obsluhou na číslicovo riadené obrábacie stroje). Postupne sú tak vytvárané riadiace systémy buď plno automatické (bez akejkoľvek účasti človeka na riadenie) alebo automatizované, kde človek do automaticky riadeného procesu zasahuje spôsobom, ktorý je skôr závislý na charaktere riadeného procesu (napr. modifikuje spôsob riadenia podľa okamžitého priebehu riadeného procesu) [5].

Výroba potravín v porovnaní s chemickým, farmaceutickým, či strojárenským priemyslom vykazuje malý stupeň automatizácie. Vzhľadom na cenu výroby, rastúci počet obyvateľov bude potrebné používať vyšší stupeň automatizácie v potravinárskom priemysle a to v oblasti riadenia výroby, kontroly kvality a v oblasti logistiky.[37] Pri riadení výroby sú využívané riadiace jednotky, ktoré vedú k zníženiu počtu pracovných síl (nižšia cena výrobkov) a k zvýšeniu kvality výroby.

Vývoj riadiacej jednotky sa začal s vynálezom mikrokontrolérov (čo v preklade znamená počítače integrované v jednom čipe). Mikrokontrolér obsahuje skoro všetko potrebné pre prevádzku, čiže procesorové jadro, volatilnú a nonvolatilnú pamäť, časovače, AD/DA prevodníky, externé prerušenie, USART a mnoho iných.

Diplomová práca sa zaoberá s návrhom riadiacej jednotky pre potravinársky priemysel, ktorá bude so svojimi funkciami cenovo porovnateľná s podobnými zariadeniami, dostupnými na trhu.

## 2. Formulácia problému a ciele riešenia

Cieľom riešenia je návrh a realizácia univerzálnej riadiacej jednotky pre riadenie jednoduchých technologických procesov v potravinárskom priemysle. Riadiaca jednotka mala byť navrhnutá na "hobby" výrobu piva. Najdôležitejšou časťou pri výrobe piva z hľadiska technológie je výroba mladiny, ktorá prebieha v troch cykloch:

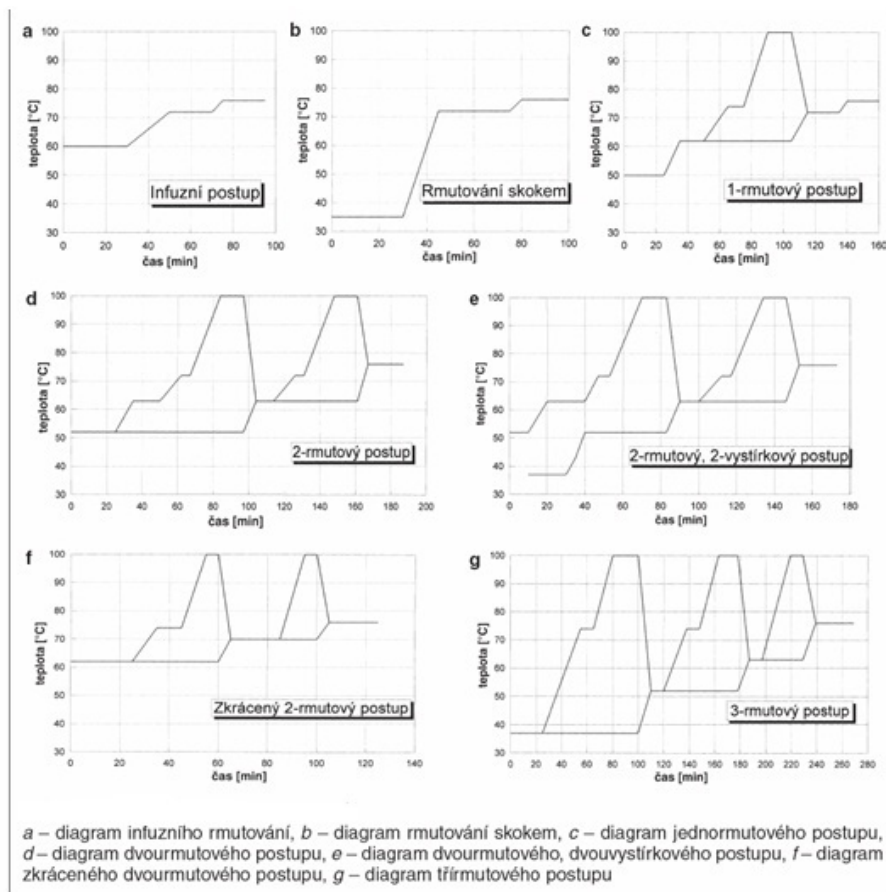
1. Rmutovanie
  - prevod škrobnatých látok sladu do roztoku
2. Cedenie
  - získavanie sladiny filtráciou scukornateného rmutu
3. Chmeľovar
  - získavanie mladiny varením sladiny s chmeľom

Veľmi dôležitou časťou pri výrobe piva z hľadiska chuti je rmutovanie. Možné postupy rmutovania vidíme na obr. č. 2.1. Rmutovací postup spočíva v rozdelení objemu mladiny do dvoch nádob a po určitom časovom úseku v ich spojení. pre podrobnejšie informácie o technológii výroby viď [7] [4].

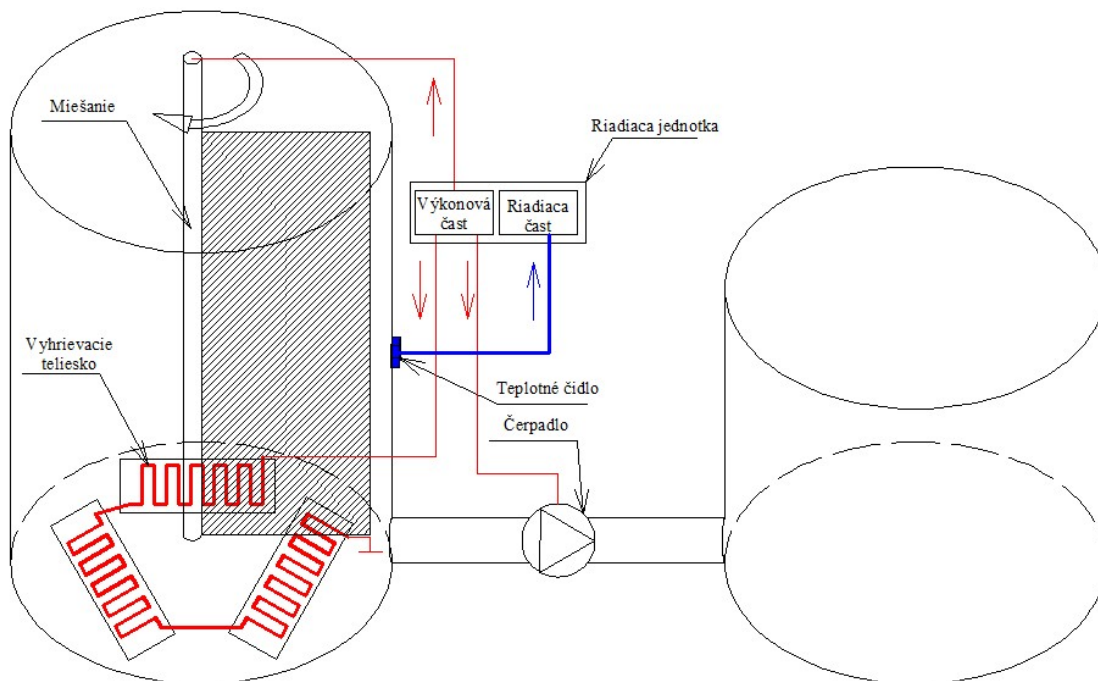
Na danú technológiu je potrebné mať riadiacu jednotku so vstupom pre snímanie teploty a s výstupom pre riadenie jednotlivých technologických operácií, ako je výkonové riadenie odporovej záťaže - ohrevu a relé pre spínanie výstupu, v našom prípade je to čerpadlo na prečerpávanie média a motor napr. na zapínanie a vypínanie miešania. Odporová záťaž je reprezentovaná výhrevnou špirálou, ktorá je napájaná na striedavé napätie a má výkon 2800 W. Schéma zariadenia na výrobu piva je znázornená na obr. č. 2.2.

Priebehy žiadaných veličín ako sú priebehy rmutovacích postupov je potrebné naprogramovať na osobnom počítači a následne ich prekopírovať do riadiacej jednotky. Tento postup nám umožňuje jednoduchú zámenu rôznych známych a používaných spôsobov varenia piva. Po nahraní veličín by mala riadiaca jednotka pracovať v autonómnom režime. Základná bloková schéma riadiacej jednotky je znázornená na obr.č. 2.3.

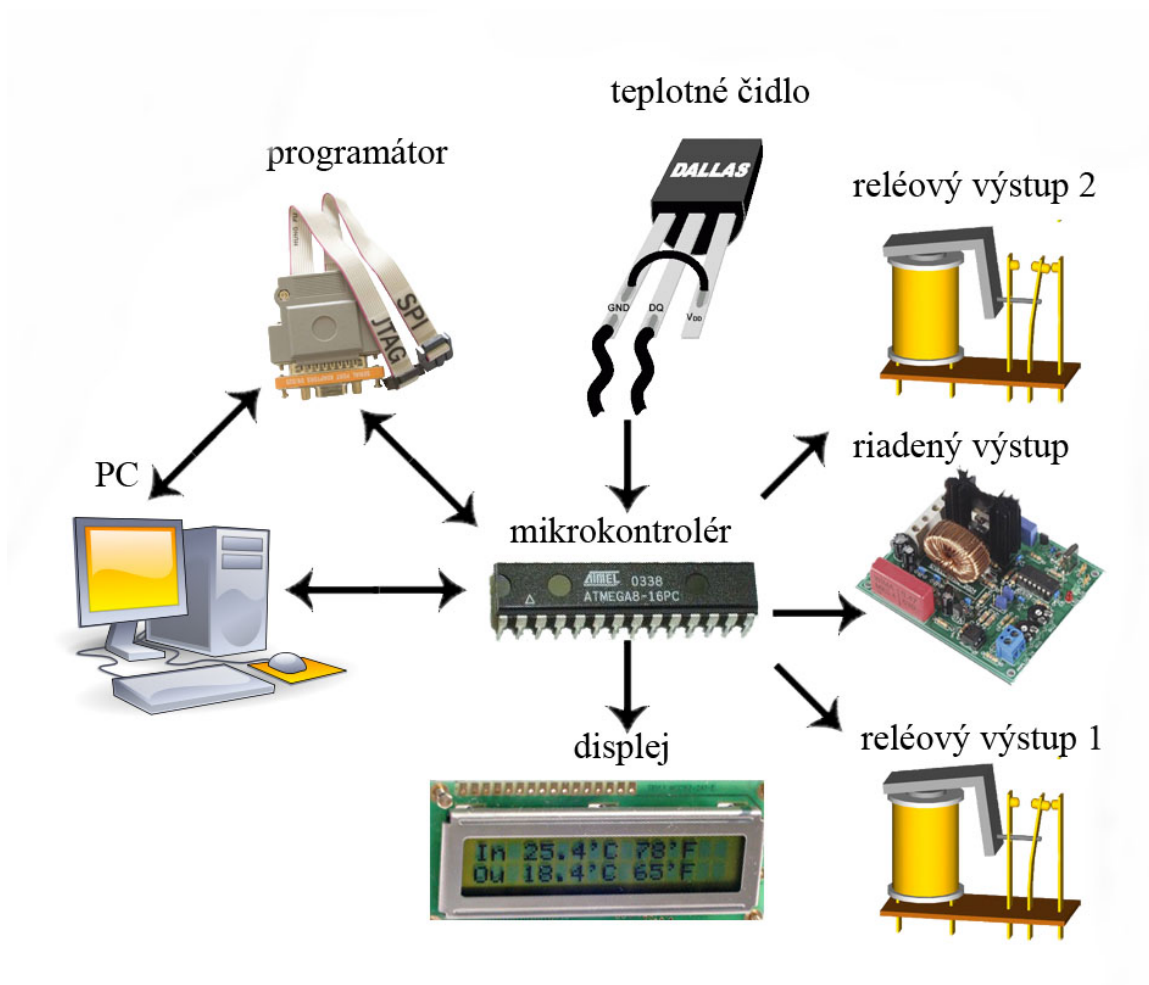
## 2. FORMULÁCIA PROBLÉMU A CIELE RIEŠENIA



Obr. 2.1: Príklady rmutovacích postupov [4]



Obr. 2.2: Základná schéma zariadenia na výrobu piva



Obr. 2.3: Základná schéma riadiacej jednotky

Pred návrhom sústavy prevedieme rešeršné štúdium o podobných riadiacich jednotkách, dostupných na trhu. Na základe týchto faktov boli ustanovené nasledovné ciele diplomovej práce:

1. Navrhnuť riadiacu jednotku na riadenie technologického procesu s presne definovanými vstupmi a výstupmi
2. Vytvoriť a odladiť softwarovú aplikáciu pre nastavenie a kontrolu parametrov riadeného procesu
3. Realizácia prototypu riadiacej jednotky

## 3. Rešeršné štúdium

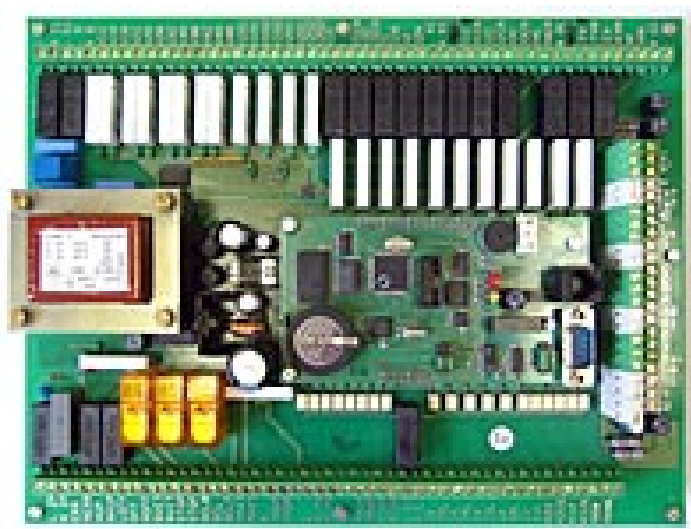
V tejto časti sú uvedené niektoré podobné riadiace jednotky so základnými vlastnosťami, ktoré sú dostupné na trhu.

### 3.1. Riadiace jednotky

#### Elitronic

Firma Elitronic [8] sa zaoberá vývojom a výrobou elektronických a elektrických zariadení. Na technológiu varenia piva by sa dal použiť univerzálny riadiaci terminál **DistEl RG340** [8], viď obr. 3.1.

RG340 je jednodoskový regulátor obsahujúci všetky základné funkčné bloky pre reguláciu ohrievaných sústav.



Obr. 3.1: Univerzálny riadiaci terminál DistEl RG340 [8]

Parametre riadiacej jednotky sú nasledovné:

- 5 univerzálnych digitálnych vstupov
- teplotné (odporové) s rozsahom ( $-50$  až  $140$ ) $^{\circ}\text{C}$ 
  - 6 vstupov pre meranie teplôt
- napäťové  $0\text{V}$  až  $10\text{V}$ 
  - 2 napäťové
- 14 digitálnych výstupov

### 3.1. RIADIACE JEDNOTKY

- jeden analógový výstup s rozsahom 0V až 10V
- rozhranie RS232 alebo RS485 na pripojenie PC

Na riadiacu jednotku je potrebné pripojiť snímač teploty a regulátor odporovej záťaže riadeného pomocou analógového výstupu.

#### MRK

**Programovateľný regulátor RR** od firmy **MRK**<sup>[10]</sup> vid'. na obr. 3.2 je vhodný pre jednoduché i veľmi náročné aplikácie s užívateľsky priateľským ovládaním. Reguluje tiež výkon elektrických ohrievacích telies podľa požadovaného priebehu teploty, ktorý je zložený z časových intervalov a žiadanej teploty vytvorenej užívateľom. Štandardne môže byť v pamäti intervalov od 1 min. do 24 hod. uložených naraz 7 programov, s dĺžkami od 1 min. do 24 hod.



Obr. 3.2: Programovateľný regulátor RR <sup>[10]</sup>

Vstupný signál :

- 1 ľubovoľný snímač
- snímanie výpadku fázy
- 1 ľubovoľný snímač

Výstupný signál :

- riadiace signály pre výkonové akčné prvky



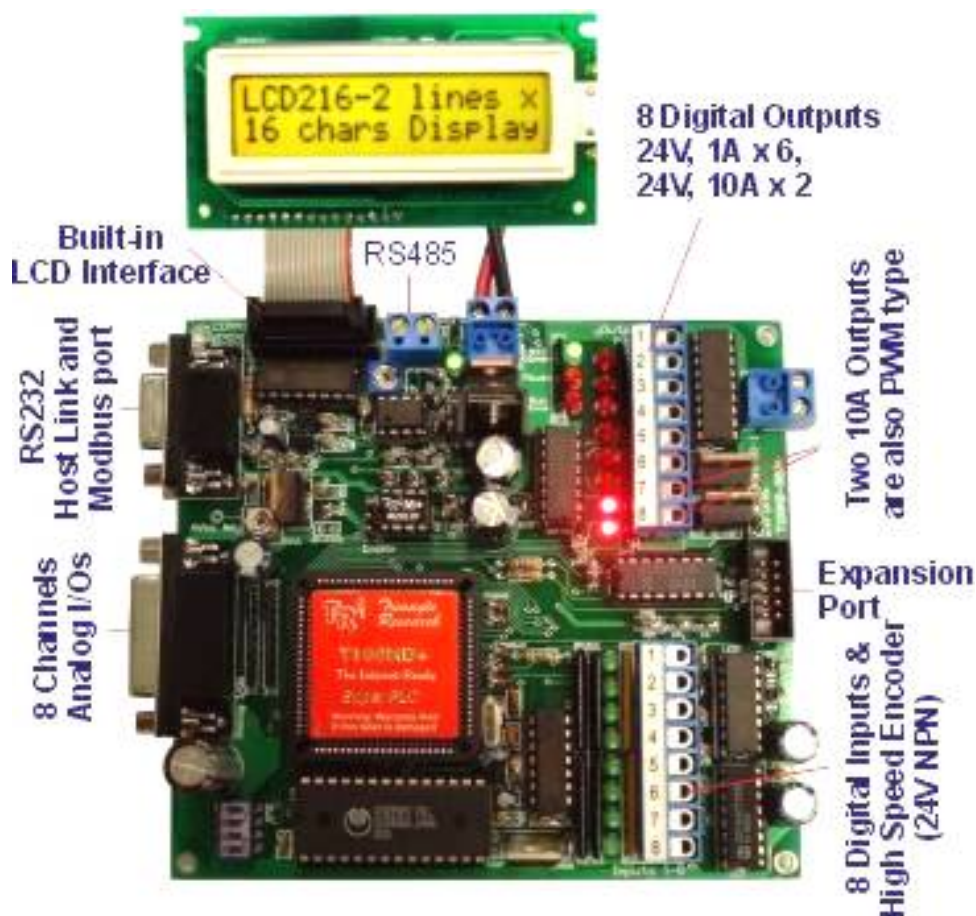
- relé 230V 8A pre ovládanie ochranného stykača
- 12V 0,08A pre ventilátor chladenia výkonového prvku

Voliteľné výstupy :

- výstupy 230V 8A pre ovládanie serva
- snímanie a regulácia ďalších kanálov
- pripojenie k PC pomocou RS 232, RS 485

### TRI-INC

Ďalšia riadiaca jednotka je typu **T100MD888+** od firmy Triangle Research International [9] vid'. obr.3.3.



Obr. 3.3: Univerzálny riadiaci terminál DistEl RG340 [9]

Parametre riadiacej jednotky:

- 8 digitálnych vstupov, z ktoré je možné použiť na prerušenia(max 4), a na enkóder

### 3.1. RIADIACE JEDNOTKY

- 6 10bitových vstupov s rozsahom 0 V až 5 V.
- 8 digitálnych výstupov vrátane PWM
- 2 10bitové analógové výstupy s rozsahom 0 V až 5 V.
- komunikačne rozhranie RS232 alebo RS485 na pripojenie PC
- LCD displej

Cena:

- 195\$ približne 3790Kč

### FATEK PLC

Firma FATEK PLC [11] má vo svojej ponuke kontrolér typu **FBs-10MAT**[11], vid' obr. 3.4, ktorého vlastnosti sú nasledovné:

- 4 vstupy pre frekvenciu 10kHz
- 2 vstupy do  $470\mu s$
- 4 reléové, alebo tranzistorové výstupy
- RS232 Port alebo USB
- Galvanicky oddelené vstupy a výstupy

Cena tejto jednotky činí 2737Kč bez DPH, vid'. [11]. Na riadiacu jednotku je potrebné pripojiť snímač teploty a regulátor odporovej záťaže riadeného pomocou digitálneho výstupu.

### Unitronics Ltd.

Spoločnosť Unitronics Ltd. sa zaoberá vývojom a výrobou programovacích automatov s označením OPLC<sup>TM</sup>. Z uvedených prevedení [14] [15] je pre nás dostačujúci typ **JZ10-11-T10**, vid' obr. 3.5, ktorého vlastnosti sú nasledovné:

- 6 digitálnych vstupov
- 4 tranzistorové výstupy



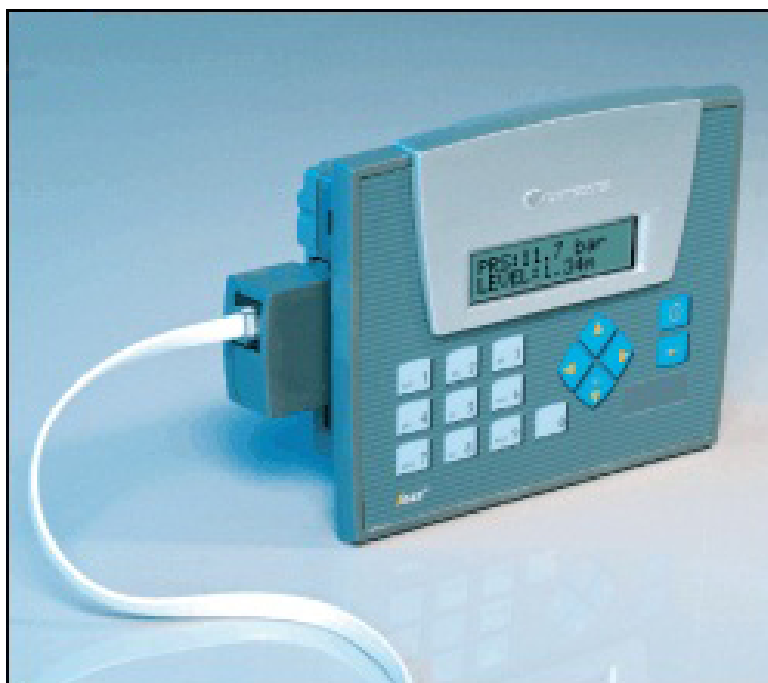
Obr. 3.4: Univerzálny riadiaci terminál FAC-FBS-10MAT [11]

- komunikačný port RS232
- dvojriadkový LCD display
- klávesnica
- 2 čítače do 5kHz

Cena tejto jednotky činí 5017 Kč (príplatok za komunikačný adaptér), viď. [14]. Na riadiacu jednotku je potrebné pripojiť triakový regulátor na riadenie odporovej záťaže, riaditeľného pomocou tranzistorového výstupu, 2 reléové výstupy a teplotné čidlo.

V našom prípade je potrebné riadiť odporovú záťaž napojenú na striedavé napätie. Na riadenie takejto záťaže je možné použiť triakový jednofázový regulátor výkonu EFM [16], viď obr. 3.6. Regulátor je proporcionálne riadený napätím (0-10)Vdc. Cena regulátora sa pohybuje okolo 2560 Kč [17].

### 3.1. RIADIACE JEDNOTKY



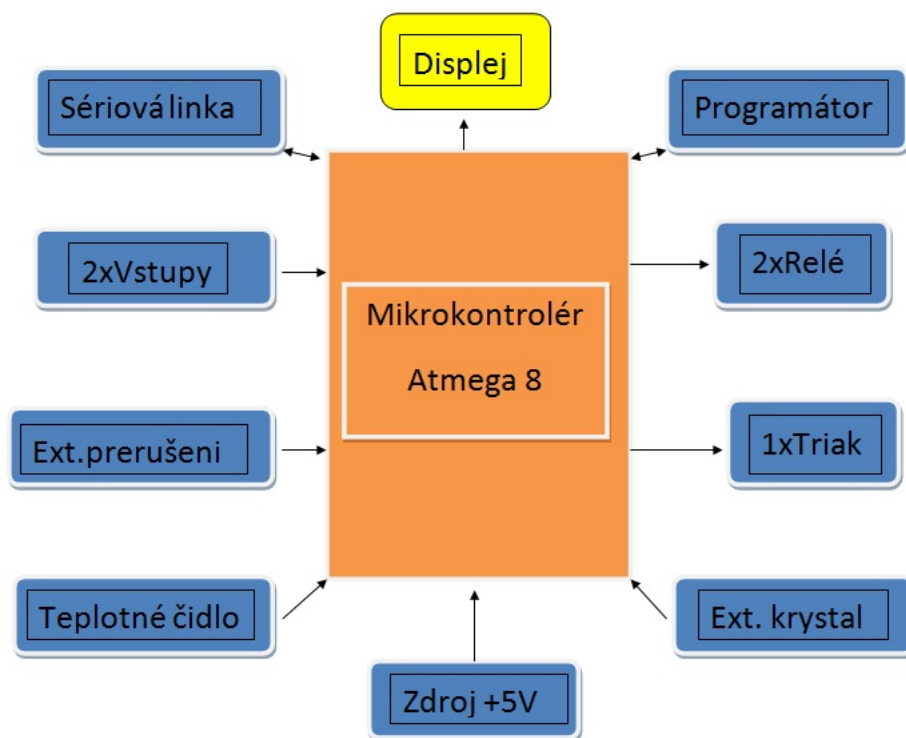
Obr. 3.5: Univerzálny riadiaci terminál FAC-FBS-10MAT [15]



Obr. 3.6: triakový jednofázový regulátor výkonu EFM [17]

## 4. Analýza problému

Každá riadiaca jednotka obsahuje výkonovú a riadiacu časť, ktoré sú od seba galvanicky oddelené. Galvanické oddelovanie slúži na prenos elektrickej energie medzi obvodmi, ktoré nie sú prepojené vodičmi. Na galvanické oddelovanie sa najčastejšie používajú transformátory a optočleny. Najdôležitejšou časťou riadiacej jednotky sú riadiaca časť, vstupná časť a výstupná časť. Riadiacu časť tvorí mikročip, ktorého súčasťou je software slúžiaci na ovládanie programu a reguláciu výstupu vzhľadom na vstup. Výstupná časť môže byť tvorená napr. spínacími prvkami, prípadne LCD displejom. Vstupná časť môže byť tvorená napr. tlačítkami, snímačmi, prípadne sériovým rozhraním. Na obr. 4.1) vidíme základnú blokovú schému riadiacej jednotky.



Obr. 4.1: Základné bloková schéma

### 4.1. Riadiaca časť

#### 4.1.1. Mikročip

Jednočip je integrácia mikroprocesora, pamäte, časovača a I/O obvodov do jedného čipu. Väčšinou je pritom oddelená pamäť programu od pamäte pre dáta - **Harvardská architektúra**. Ďalšia architektúra, ktorú počítače využívajú, je **von Neumanova architektúra**.

#### 4.1. RIADIACA ČASŤ

**túra.** Táto koncepcia využíva spoločnú pamäť pre program a dáta.

Medzi 8 bitovými jednočipmi sú u nás najpoužívannejšie nasledovné rodiny:

- mikropočítač 8051  
pochádza z roku 1980 a je vývojovo pomerne starý, ale obľúbený jednočip. Je to procesor s architektúrou jadra CISC.
- mikrokontrolér PIC  
Mikrokontroléry PIC od firmy Microchip sa vyznačujú jednoduchou štruktúrou a malým množstvom inštrukcií. Tieto mikrokontroléry sú vhodné hlavne pre jednoduché aplikácie, ich výhodou je nižšia spotreba pri menších pracovných frekvenciách a možnosť využitia bezplatného vývojového programu.
- AVR  
Mikrokontroléry AVR od firmy Atmel patria medzi najmodernejšie a veľmi obľúbené mikrokontroléry vďaka pomerne vysokému výkonu, množstvu zabudovaných periférií, veľkému množstvu variánt a priaznivej cene. Ďalšou výhodou je jednoduché zhotovenie a pripojenie programátora. Je to procesor s architektúrou jadra RISC.
- a mnoho ďalších, pre viac informácií viď. [39] .

#### Architektúra mikrokontrolérov

##### Porovnanie RISC a CISC architektúry

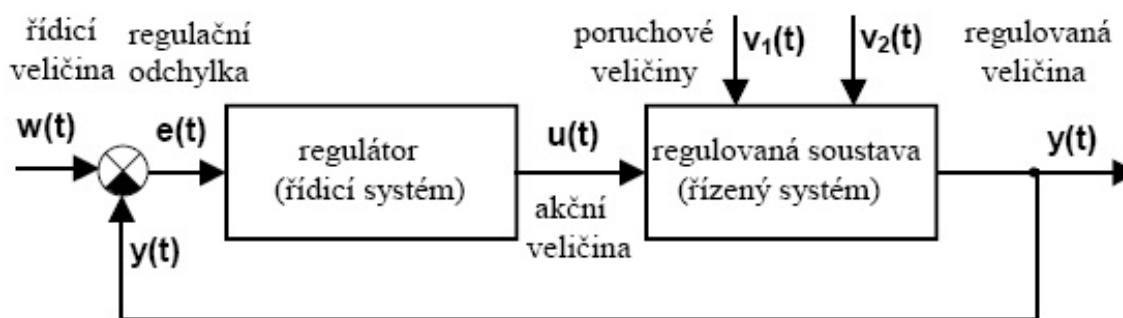
U CISC (Complex Instruction Set Computer) architektúry má inštrukčný súbor také inštrukcie, ktoré pod jedným operačným kódom vykonajú zložité operácie s variabilitou rôznych adresovacích módo. Obsluha týchto inštrukcií vyžaduje viac strojových cyklov. U RISC (Reduced Instruction Set Computer) architektúry je v porovnaní s architektúrou CISC podstatná jednoduchá štruktúra inštrukcie a najmä jeho priamočiare vykonávanie. Koncepcia RISC je zbavená niektorých zložitejších inštrukcií, ktoré sa vo väčšine programov využívali zriedkavo alebo sú nahradené postupnosťou jednoduchých inštrukcií. Výhodou architektúry RISC je zmenšenie veľkosti púzdra a zvýšenie výpočetného výkonu. Riadiace obvody u CISC architektúry zaberaajú na čipe 60% miesta a u RISC architektúry iba 6-10%. Výkon RISC architektúry okrem niektorých inštrukcií je jeden strojový cyklus. Pre viac informácií viď. [40] .

### 4.1.2. Regulácia, typy regulácie a regulátorov

#### 1. Spojité riadenie

Regulácia je udržiavanie určitej fyzikálnej veličiny na konštantnej hodnote alebo inak podľa pravidiel sa meniacej hodnote. Behom regulácie sa zisťujú hodnoty tejto veličiny a porovnávajú sa s požadovanou hodnotou. Podľa zistených odchýlok sa zasahuje do regulačného procesu v tom zmysle, aby sa odchýlky odstránili, prípadne zminimalizovali. Na obr. č. 4.2. vidíme základnú blokovú schému regulačného obvodu. Veličina, ktorej hodnota je výstupom z regulovanej sústavy a ktorá sa reguláciou udržiava na požadovanej hodnote sa nazýva *regulovaná veličina* a označuje sa symbolom  $y$ . Pomocou *riadiacej veličiny*  $w$  (nazýva sa aj *požadovaná veličina*) nastavujeme hodnotu, ktorú má dosiahnuť regulovaná veličina. V regulačnom obvode sa neustále meria hodnota regulovanej veličiny a porovnáva sa so žiadanou veličinou a tým vytvára *regulačnú odchýlku*  $e$  danú vzťahom 4.1.

$$e = w - y \quad (4.1)$$



Obr. 4.2: Základná bloková schéma regulačného obvodu [5]

Do regulačného procesu treba zasahovať tak, aby regulačná odchýlka bola minimálna, prípadne nulová. To sa uskutočňuje *akčnou veličinou*  $u$ , ktorá je vstupnou veličinou regulovanej sústavy a výstupnou veličinou regulátora. Regulátor môže regulačnú odchýlku zosilovať, integrovať a derivovať. Podľa toho môžeme rozdeliť regulátory na nasledovné základné typy:

- Proporcionálne (P regulátor)
- Integročné (I regulátor)
- Derivačné (D regulátor)

## 4.1. RIADIACA ČASŤ

### **P regulátor**

Je najjednoduchšia forma regulátora. Akčná veličina je priamo úmerná regulačnej odchýlke:

$$u = K_P \cdot e \quad (4.2)$$

Z toho vyplýva, že pri dostatočne veľkej regulačnej odchýlke bude akčná veličina veľká. Akonáhle bude rozdiel medzi skutočnou a požadovanou hodnotou nulový, tak aj zásah bude nulový. Vlastnosti P regulátora závisia od nastavení konštanty  $K_P$ . Ak nastavíme príliš vysokú konštantu, tak na malú regulačnú odchýlku bude reagovať veľkou akčnou veličinou a výsledkom bude kmitanie sústavy okolo požadovanej hodnoty. Ak sa nastaví malá Pkonštantu regulátor nikdy nedosiahne žiadanú veličinu. U P regulátoroch je častým problémom trvalá regulačná odchýlka po ustálení systému. P regulátor nie je vhodný pre regulované sústavy vyššieho rádu a pre sústavy s dopravným oneskorením.

### **I regulátor**

U I regulátoroch je akčná veličina úmerná regulačnej odchýlke:

$$u = K_I \cdot \int e dt \quad (4.3)$$

Zásah integračného regulátora je závislý ako aj na regulačnej odchýlke, tak aj na dobe trvania regulačnej odchýlky. Integračný regulátor zaistí minimalizáciu, prípadne nulovanie regulačnej odchýlky, čiže po ustálení je regulačná odchýlka malá, alebo nulová. I regulátor je vhodný na proporcionálne regulované sústavy a pre reguláciu statických sústav s dopravným oneskorením.

### **D regulátor**

Samotný D regulátor v praxi neexistuje, iba pri kombinácii vyššie spomínaných P a I regulátorov. Akčná veličina D zložky regulátora je úmerná derivácii regulačnej odchýlky:

$$u = K_D \frac{de}{dt} \quad (4.4)$$

Zásah derivačného regulátora je závislý od zmeny rýchlosti regulačnej odchýlky. Ak regulačná odchýlka klesá, D regulátor zvyšuje zásah, v opačnom prípade, ak sa bude odchýlka zvyšovať, zásah bude menší.

Kombináciou týchto troch základných regulátorov vzniknú ďalšie regulátory.



**PI regulátor**

Je kombináciou P a I regulátora, preto sa aj nazýva PI regulátor. PI regulátor má akčnú veličinu úmernú regulačnej odchýlke a aj jej integrálu 4.5, pričom vplyv každej zložky sa dá zmeniť zmenou konštánt  $K_P$  a  $K_I$ .

$$u = K_P \cdot e + K_I \cdot \int e dt \quad (4.5)$$

**PD regulátor**

Podobne ako PI tak aj PD regulátor je kombináciou P a D regulátora. PD regulátor má akčnú veličinu úmernú regulačnej odchýlke a aj jej derivácii 4.6, pričom vplyv každej zložky sa dá zmeniť zmenou konštánt  $K_P$  a  $K_D$ .

$$u = K_P \cdot e + K_D \frac{de}{dt} \quad (4.6)$$

PI regulátor úplne odstraňuje regulačnú odchýlku a zlepšuje stabilitu regulačného obvodu.

**PID regulátor**

Kombináciou základných troch regulátorov je Proporcionálne-integračno-derivačný PID regulátor, ktorý má akčnú veličinu úmernú regulačnej odchýlke a aj jej integrálu a derivácie 4.7, pričom vplyv každej zložky sa dá zmeniť zmenou konštánt  $K_P$ ,  $K_D$  a  $K_I$ .

$$u = K_P \cdot e + K_I \cdot \int e dt + K_D \frac{de}{dt} \quad (4.7)$$

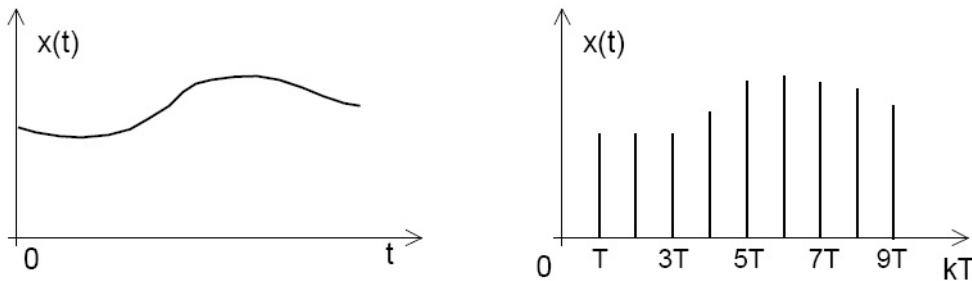
Regulátor PID je vhodný všade tam, kde sa používa PI. V porovnaní s PI regulátorom je PID rýchlejší a lepšie tlmí prekmity regulovanej veličiny.

**2. Diskrétna riadenie**

Spojité signály sú spojité funkcie času, t.j., v každom časovom okamžiku môžeme určiť hodnotu tohto signálu. Vzorkovaním spojitých signálov v určitých časových intervaloch dostaneme diskretný signál, viď. obr. č. 4.3.

Príchodom počítačov sa začali regulátory zhotovovať softvérové, kde spojitý (hardvérový) PID regulátor začal byť nahradený číslicovým (softvérovým) regulátorom. Počítač zisťuje vstupnú veličinu v určitých časových okamihoch a v tom časovom okamihu vypočíta akčnú veličinu. Diskrétna riadenie sa v porovnaní so spojitým vyznačuje vyššou statickou dlhodobou presnosťou, menšou citlivosťou na elektromagnetické rušenie a menšou necitlivosťou ku zmene teploty.

#### 4.1. RIADIACA ČASŤ



Obr. 4.3: Priebehy spojitého a diskrétného signálu v závislosti na čase[6]

#### Číslicové regulátory

Od číslicových regulátorov budeme očakávať rovnakú funkciu ako od spojitých regulátorov, t.j. vstupnú veličinu budeme zosilovať, integrovať a derivovať. Preto pri zostavovaní rovníc číslicového PID regulátora budeme vychádzať zo spojitého PID. Rovnica spojitého PID regulátora má tvar:

$$u(t) = K_p \left( e(t) + \frac{1}{T_I} \int_0^t e(t) dt + T_D \frac{de(t)}{dt} \right) \quad (4.8)$$

Číslicovú formu regulátora získame z rovnice 4.8 diskretizáciou integrácie a derivácie. Integráciu prevedieme náhradou spojitého signálu stupňovitou obdĺžnikovou náhradou zľava. Hodnota integrálu bude súčet všetkých plôch pod náhradným priebehom vid'. obr. č. 4.4. Na výpočet integrálu dostávame nasledovný vzťah:

$$\int_0^{kT} e(t) dt \approx T \sum_{i=1}^k e(i) \quad (4.9)$$

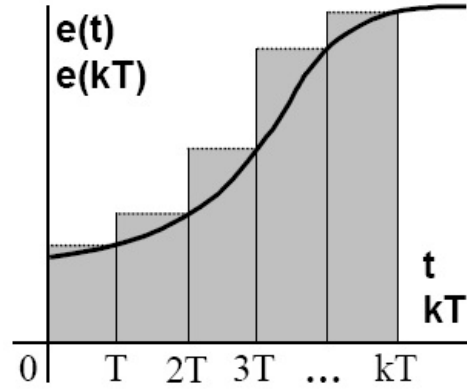
Deriváciu získame náhradou diferenciami podľa obr. č. 4.5 a získame rovnicu:

$$\frac{de}{dt} \approx \frac{e(k) - e(k-1)}{T} \quad (4.10)$$

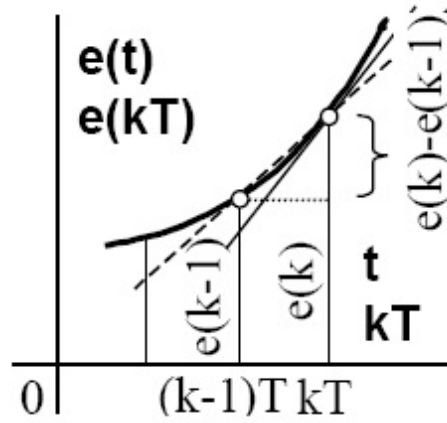
Po dosadení rovníc 4.10 a 4.9 do rovnice spojitého PID regulátora 4.8 dostaneme rovnicu:

$$u(k) = K_p \left( e(k) + \frac{T}{T_I} \sum_{i=1}^k e(i) + \frac{T_D}{T} (e(k) - e(k-1)) \right) \quad (4.11)$$

Po vynásobení konštantou  $K_p$  a zjednodušení dostaneme tvar, vid'. rov. 4.12



Obr. 4.4: Obdĺžniková metóda integrácie[5]



Obr. 4.5: Náhrada derivácie diferenciami[5]

$$u(k) = P \cdot e(k) + I \sum_{i=1}^k e(i) + D(e(k) - e(k-1)) \quad (4.12)$$

kde:

$$\begin{aligned} P &= K_p \\ I &= \frac{K_p \cdot T}{T_I} \\ D &= \frac{K_p \cdot T_d}{T} \end{aligned}$$

Rovnica 4.12 je totožná s rovnicou, ktorú uvádza literatúra [23]. Uvedenú rovnicu sme implementovali do jazyka C a použili na reguláciu. Podrobnejšie odvodenie rovníc a popis funkcie regulátora nájdeme v [5] alebo [6].

## 4.1. RIADIACA ČASŤ

### 4.1.3. Možnosti regulácie odporovej záťaže

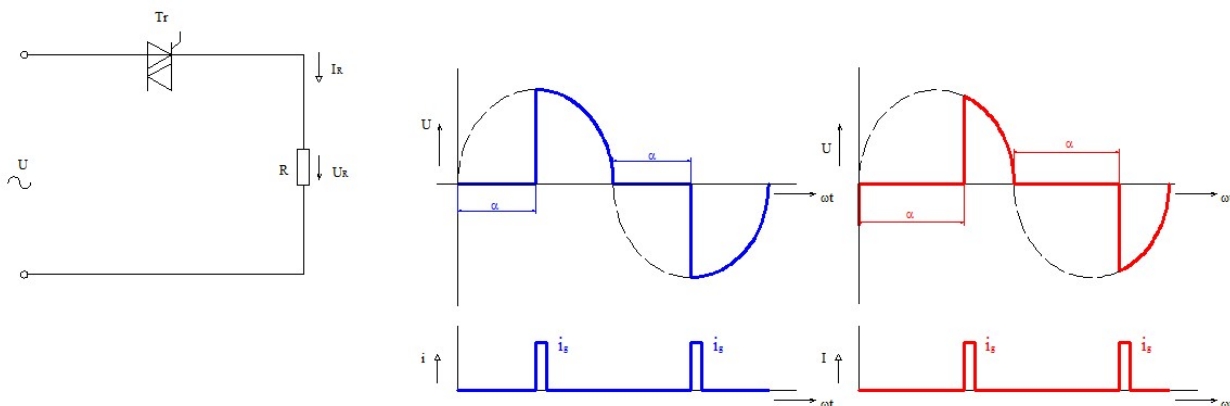
Výstupný výkon striedavého napätia pre odporovú záťaž môžeme regulovať zmenou efektívnej hodnoty striedavého napätia. K tomu je vhodné použiť striedavé meniče napätia, ktoré zachovávajú frekvenciu a menia efektívnu hodnotu napätia a tým pádom menia aj výstupný výkon. Striedavé meniče napätia poznáme:

- Jednofázový menič napätia
- Cyklovo riadený spínač
- Striedače

#### Jednofázový striedavý menič napätia

Odporová záťaž je napojená na striedavé napätie cez triak. Zmenou riadiaceho uhla triaka  $\alpha$  môžeme meniť efektívnu hodnotu výstupného napätia podľa vzťahu 4.13. Schému zapojenia a priebehy napätia v závislosti na zmene riadiaceho uhla na odporovej záťaži vidíme na obr. č. .

$$U_R = U \sqrt{1 - \frac{\alpha}{\pi} + \frac{\sin 2\alpha}{2\pi}} \quad (4.13)$$



Obr. 4.6: Schéma zapojenia a priebehy napätia na striedavom meniči napätia

### Cyklovo riadený spínač

Princíp riadenia je podobný ako v predchádzajúcom prípade. Hlavným rozdielom je hodnota riadiaceho uhla  $\alpha$ , ktorý môže nabývať hodnotu buď  $0^\circ$ , alebo  $180^\circ$ , t.j. buď je triak úplne otvorený alebo zatvorený. U cyklovo riadených spínačov môžeme vypočítať efektívnu hodnotu výstupného napätia na odporovej záťaži podľa vzťahu 4.15.

$$U_R = U \sqrt{\frac{t_{on}}{t}} \quad (4.14)$$

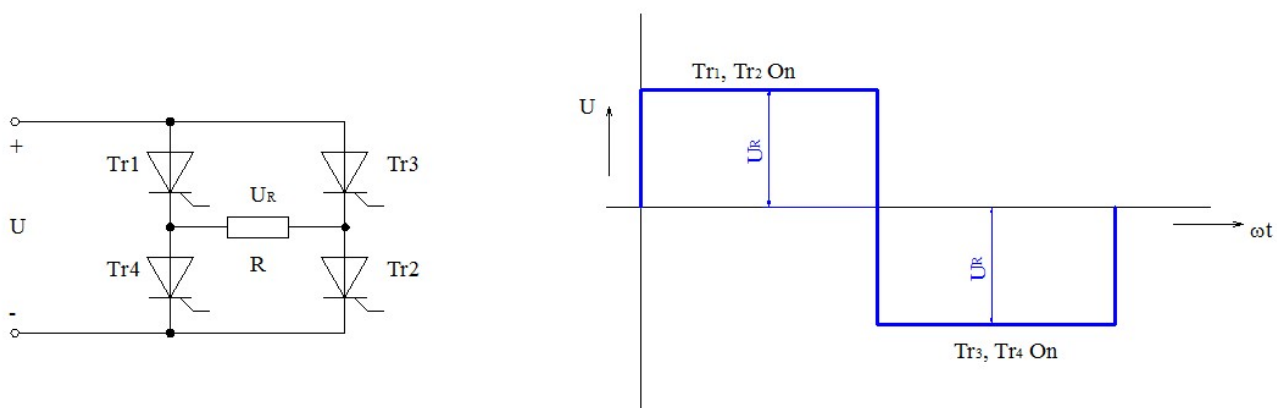
$$(4.15)$$

kde:

$t_{on}$  je doba otvorenia meniča  
 $t$  je celkový čas cyklu

### Striedač

V tomto prípade prepíname jednosmerné napätie na striedavé pomocou štyroch tyristorov. Schému zapojenia priebehu napätia na odporovej záťaži vidíme na obr. č. . Keď je zapnutý tyristor Tr.1 a Tr.2 na výstupe bude napätie  $+U$ . V prípade, ak chceme mať na výstupe  $-U$  musíme zapnúť tyristory Tr.3 a Tr.4.



Obr. 4.7: Schéma zapojenia a priebehy napätia na striedači

## 4.1. RIADIACA ČASŤ

### 4.1.4. Snímače teploty

Na reguláciu pri riadení výstupu je potrebné pripojiť na vstup vhodný snímač fyzikálnych veličín. V našom prípade je to snímač teploty. V priemysle patrí teplota medzi najčastejšie sledované a regulované fyzikálne veličiny. Teplotu môžeme merať rôznymi spôsobmi. Snímače teploty v priemysle so zameraním na reguláciu môžeme rozdeliť podľa:

- spôsobu merania
  - Dotykové - elektrické, dilatačné, tlakové
  - Bezdotykové - pyrometre
  - Špeciálne
    - \* Kryštálové (závislosť rezonančnej frekvencie kryštálového rezonátora na teplote),
    - \* Dilatačné (využívajú tepelnú rozťažnosť materiálu)
    - \* Akustické
- výstupu
  - Analógový výstupný signál
  - Digitálny výstupný signál

Medzi dotykové elektrické snímače patria:

- Odporové kovové
- Odporové polovodičové
- Polovodičové s PN prechodom
- Termoelektrické

**Odporové snímače teploty** patria medzi dotykové snímače. Ich čidlo prevádza tepelnú zmenu prostredia na zmenu elektrického odporu.

Odporové snímače teploty môžeme rozdeliť na:

- Kovové
  - Platinové
    - \* pracovný rozsah od  $(-200 \text{ do } 850)^{\circ}\text{C}$
    - \* čistota platiny ovplyvňuje statickú charakteristiku
  - Niklové
    - \* pracovný rozsah od  $(-200 \text{ do } 200)^{\circ}\text{C}$
    - \* V porovnaní s platinovým majú nižšiu cenu, vyššiu citlivosť a rýchlu časovú odozvu.
- Polovodičové
  - Pozistory
  - Negastory
  - Monokryštalické

Podobne ako u kovových aj u polovodičových snímačov teploty sa využíva závislosť elektrického odporu na teplote.

U negastorov (NTC (Negative Temperature Coefficient)) s rastúcou teplotou rastie koncentrácia voľných nosičov náboja a tým klesá elektrický odpor. Vyznačujú sa nelineárnou závislosťou odporu na teplote. Pozistory (PTC (Positive Temperature Coefficient)) sa vyznačujú prevažne kladným teplotným koeficientom a nelineárnou závislosťou odporu na teplote. Pri meraní teploty sa väčšinou používajú ako dvojstavové detektory. Z monokryštalických snímačov sa v praxi väčšinou používajú kremíkové. Vyznačujú sa dlhou stabilitou a relatívne malou nelinearitou.

Polovodičové snímače teploty majú nasledovné výhody:

- Jednoduchá výroba, snímač a elektronika je umiestnená v jednom púzdre
- Veľmi často sú vyrábané ako "SMART" snímače

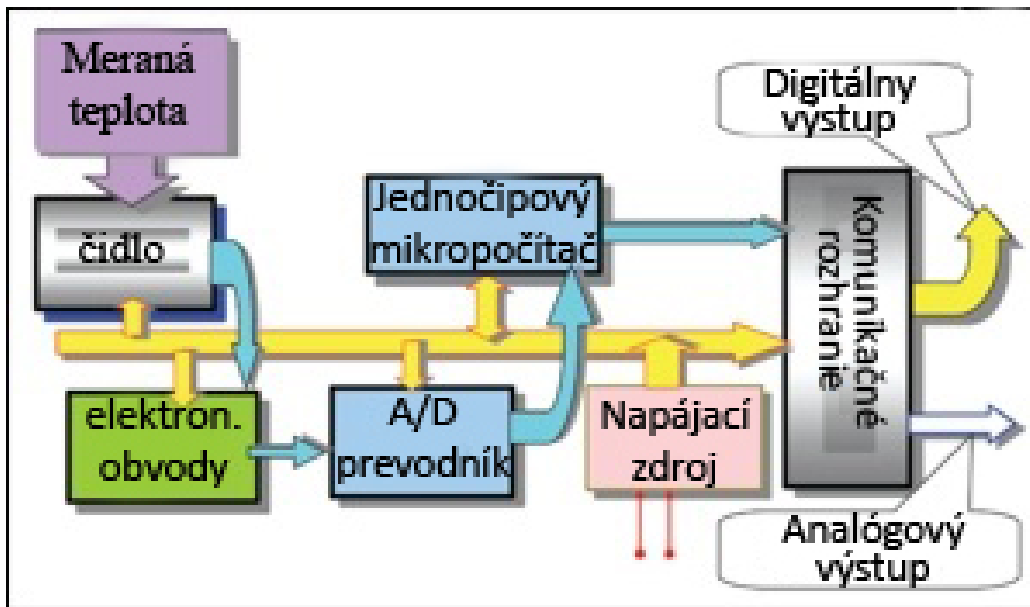
Nevýhody polovodičových snímačov teploty:

- Malý pracovný rozsah teplôt od  $(-200 \text{ do } 150)^{\circ}\text{C}$

#### 4.1. RIADIACA ČASŤ

- Niektoré typy majú silne nelineárne statické charakteristiky

Špeciálnou skupinou u polovodičových snímačov sú tzv. SMART (nazývané aj ako inteligentné) snímače. Vyrábajú sa podobným spôsobom ako integrované obvody. U týchto snímačov sa v jednom púzde nachádza samotný snímač teploty, A/D prevodník a mikro počítač, viď obr. č. 4.8



Obr. 4.8: Blokový diagram inteligentného čidla [24]

Výhodou týchto snímačov sú malé rozmery, ľahko spracovateľný výstupný signál, jednoduchá komunikácia s nadradeným systémom. Ich nevýhodou v porovnaní s ostatnými snímačmi je odklon nameranej veličiny od skutočnej veličiny (dané vlastnosťami kremíkových snímačov), malý pracovný rozsah ( $-80$  do  $120$ ) $^{\circ}\text{C}$  a pomalá časová odozva. Najčastejšie používané typy inteligentných snímačov sú TMP03 (Analog Devices), SMT160-30 (Smartec), DS18B20 (Dallas Semiconductors) a iné.

**Polovodičové snímače teploty s PN prechodom** využívajú teplotnú závislosť PN prechodu v priepustnom smere. Pracovný rozsah teplôt u týchto snímačov je od  $(-55$  do  $150)^{\circ}\text{C}$ . Môžeme ich rozdeliť na:

- Malý pracovný rozsah teplôt od  $(-200$  do  $150)^{\circ}\text{C}$
- Niektoré typy majú silne nelineárne statické charakteristiky



**Termoelektrické články** využívajú Seebeckov jav <sup>1</sup>, čiže priamo prevádzajú tepelnú energiu na elektrickú. Termoelektrické články merajú teplotu relatívne, čiže pri meraní je potrebné zaistiť, aby porovnávací spoj mal konštantnú teplotu.

Vzhľadom na výhody a nevýhody jednotlivých snímačov je pre naše použitie výhodné použiť inteligentný snímač DS18B20. Uvedený snímač má dostatočnú presnosť na použitie regulácie, jeho ďalšou výhodou je jednovodičová a jednoduchá komunikácia. Použitie a popis činnosti sa nachádza v kap. 5.1.4.

---

<sup>1</sup>Seebeckov jav je teda vznik napätia, ktorý nastáva pri teplotných rozdieloch medzi dvoma rozdielnymi kovmi alebo polovodičmi. To spôsobuje nepretržité prúdenie elektrónov, pokiaľ vodiče tvoria uzatvorený obvod. Vzniknuté napätie je rádovo niekoľko mikrovoltov na stupeň Celzia.[35]

# 5. Realizácia riadiacej jednotky

## 5.1. Riadiaca časť

### 5.1.1. Mikrokontrolér Atmel AVR Atmega 8

Z hľadiska ceny a výkonu je na našu aplikáciu vhodná Atmega 8 od firmy Atmel [41]. Ako už bolo spomenuté, všetky mikrokontroléry AVR sa vyznačujú RISC architektúrou, veľkým počtom pracovných registrov a sériovým downloadom, čo nám umožňuje sériové programovanie (pri paralelnom programovaní je potrebné premiestňovať mikrokontrolér do programátora a po nahraní programu zase naspäť do aplikačnej dosky).

Atmega 8 je nízkopríkonový 8bitový mikrokontrolér s výpočtovým výkonom 1MIPS (Mega Instructions Per Second) na 1MHz, čiže pri frekvencii 8MHz výpočtový výkon dosahuje hodnotu 8MIPS. Inštrukčný súbor obsahuje 130 inštrukcií a má k dispozícii 32 registrov s dĺžkou 8bitov. Mikrokontroléry AVR majú Harvardskú architektúru (rozdelená pamäť programu a pamäť dát). Pre pamäť programu je používaná Flash pamäť s kapacitou 8KB a so životnosťou 10000 zápisov a pre pamäť dát 1KB SRAM a 512B EEPROM so životnosťou 100 000 zápisov.

Súčasťou mikrokontrolérov sú periférie, ktoré rozširujú ich vlastnosti a slúžia na komunikáciu s okolím. Periférie sú samostatné časti mikroprocesora, nezávislé na prebiehajúcich inštrukciách. Periférie Atmegy 8 sú nasledovné:

- dva 8 bitové čítače/časovače
- jeden 16 bitový čítač/časovač
- tri kanály pre PWM
- USART-asynchrónny sériový kanál
- A/D prevodník
- SPI (Serial Peripheral Interface)
- Watchdog Timer
- 28 programovateľných vstupno-výstupných portov

## Čítače/časovače

Čítače slúžia na čítanie impulzov privedené z vonkajšieho zdroja. Čítače môžeme využiť na meranie frekvencie (počet impulzov za 1s) alebo na meranie času pomocou pravidelne prechádzajúcich impulzov od známej frekvencie.

V režime časovača sa číta pevná frekvencia, ktorá je odvodená od frekvencie mikroprocesora. Čítacia frekvencia môže byť nastavená predeličkou.

Čítače/časovače umožňujú časovať vykonávanie programu, meranie časových parametrov signálov, generovanie PWM (Pulse Width Modulation) signálu (celkovo 14 módov) a pod. Súčasťou čítača/časovača je záchytný register a výstupný komparátor. Záchytný register umožňuje zachytiť stav časovača v zvláštnom registri. K zachyteniu môže dôjsť buď aktiváciou vstupu ICP alebo pomocou zabudovaného analógového komparátora. Register výstupného komparátora obsahuje hodnoty, ktoré porovnáva s obsahom čítača a pri ich zhode môže byť vyvolané prerušenie, alebo zmena generovania výstupu.

V našom prípade bol využitý čítač/časovač 0 na meranie časového úseku pri dosiahnutí požadovanej hodnoty a pri odosielaní nameraných hodnôt pri meraní. Pri meraní časového úseku po dosiahnutí žiadanej veličiny použijeme interval 1s. Nastavením  $TCCR0=0x05$  nastavíme predeličku na 1024, tým dosiahneme vstupný kmitočet  $8000000/1024 = 7812,5\text{Hz}$ . Pre dosiahnutie frekvencie 1Hz potrebujeme, aby k pretečeniu časovača došlo po 7812,5 pulzoch. Keďže čítač je 8 bitový,  $7812,5/250 = 31,25$  budeme musieť počítať každé 31. prerušenie. Na začiatku programu sme nastavili povolenie prerušenia po pretečení čítača/časovača 0 nastavením  $TIMSK=0x01$ . Podrobnejšie informácie viď. [41].

## Generovanie PWM

Na generovanie PWM signálu môžeme vybrať z nasledujúcich 5 režimov a celkovo ich môžeme rozdeliť do 14 módov.

- Normálny režim

Pri tomto režime čítač/časovač číta hore a zvyšuje obsah čítača o 1. Pri dosiahnutí maxima dojde k nulovaniu čítača.

- CTC (Clear Timer on Compare Match)

Pri tomto režime dechádza k vynulovaniu čítača, keď hodnota čítača/časovača TCNT1 dosiahne zhodu s hodnotou výstupného komparátora (OCR1) alebo s hodnotou záchytného registra (ICR1). Takže OCR1 alebo ICR1 (výber závisí na móde) v tomto

## 5.1. RIADIACA ČASŤ

režime nastavuje vrchol čítača/časovača 1 a po dosiahnutí vrchola môže dôjsť k prerušeniu.

- Fázovo korigovaný PWM režim

Pri tomto režime je čítač inkrementovaný pokiaľ nedosiahne pevne nastavené hodnoty  $0xFF$ ,  $0x1FF$ ,  $0x3FF$  alebo hodnoty ICR1 alebo OCR1. Po dosiahnutí nastavenej hodnoty čítač zmení smer čítania. Výber týchto hodnôt, kedy dosiahne čítač vrchol závisí od módu.

- Fázovo a kmitočtovo korigovaný PWM režim

Pri tomto režime podobne ako u predchádzajúceho režimu je obsah čítača inkrementovaný pokiaľ nedosiahne vrchol, ktorý je definovaný buď pomocou registra ICR1 alebo pomocou registra OCR1. Rozdiel medzi fázovo a kmitočtovo korigovaným PWM režimom a predchádzajúcim je v okamihu, keď je register OCR1 aktualizovaný zo svojho buffera.

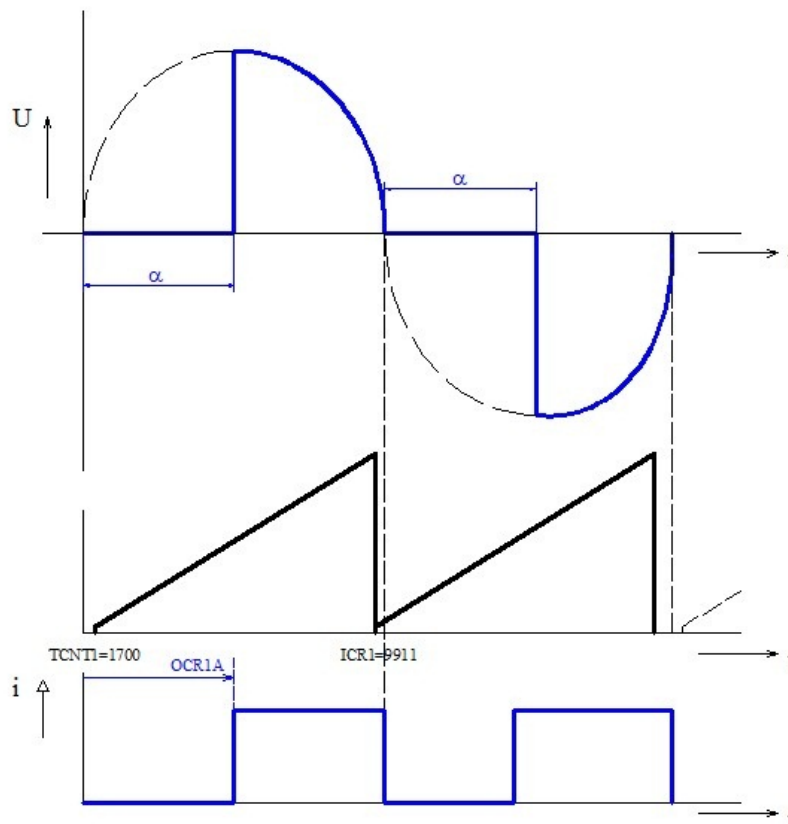
- Rýchly PWM režim

Pri tomto režime čítač číta od dna až k vrcholu, ktorý je definovaný buď pevne nastavenou hodnotou  $0xFF$ ,  $0x1FF$ ,  $0x3FF$  alebo hodnotou ICR1 alebo OCR1. Po dosiahnutí vrchola dôjde k vynulovaniu obsahu čítača a znova začína čítať od dna.

Pre podrobnejší popis viď. [1], prípadne [41]. V našom prípade sme použili na generovanie PWM signálu čítač/časovač 1, kde sme vybrali rýchly PWM režim a 14. mód. Generovaný PWM signál slúži na nastavenie riadiaceho uhla triaka viď. kap. 4.1.3. Pri používaní PWM signálu u striedavého napätia musíme zosynchronizovať čítač/časovač so sieťovou frekvenciou a správne nastaviť parametre čítača/časovača1. V našom prípade sme na synchronizáciu použili optočlen, viď. 10.3. Keďže diódou prechádzajú iba kladné napätia, optočlen sníma každú kladnú časť nábežnej hrany sieťovej frekvencie. Zostupná hrana prechodu nulou vyvolá externé prerušenie, v ktorom sa obsah čítača vynuluje. Ďalšie nulovanie obsahu čítača dochádza pri dosiahnutí vrchola, ktorý je v tomto prípade nastavený pomocou registra ICR1. Pretože pri jednej perióde musíme dvakrát vynulovať obsah čítača a keďže jedna polperióda pri sieťovej frekvencii 50Hz trvá  $10000ms$ , teoreticky by mala byť hodnota ICR1=10000. Z bezpečnostných dôvodov však vynulujeme čítač o niečo skôr, aby nedošlo k neskorému vynulovaniu(po priechode 0) a tým pádom

## 5. REALIZÁCIA RIADIACEJ JEDNOTKY

by zostal triak plne otvorený. V našom prípade je nastavená hodnota  $ICR1=9911$ . Do úvahy musíme zobrať aj fakt, že k vynulovaniu časovača pomocou externého prerušenia dôjde s určitým dopravným oneskorením zapríčineným neskoršou reakciou LED diódy na napätie. Na elimináciu tohto problému je potrebné nastaviť obsah registra  $TCNT1$  na hodnotu dopravného oneskorenia. V našom prípade je to hodnota  $TCNT1=1700$ , čo je vlastne dopravné oneskorenie  $1700ms$ . Keď už máme zosynchronizovaný čítač/časovač a správne nastavené nulovanie, môžeme nastavovať uhol otvorenia triaka pomocou registra  $OCR1A$ . Pri invertujúcom režime je vývod  $OC1$  trvalo nastavený na úroveň  $log.0$ . Po dosiahnutí hodnoty  $TCNT1=OCR1A$  dôjde k nastaveniu vývodu do  $log.0$  viď obr. č. 5.1.



Obr. 5.1: Nastavenie PWM pomocou časovača  $OCR1A=5000$

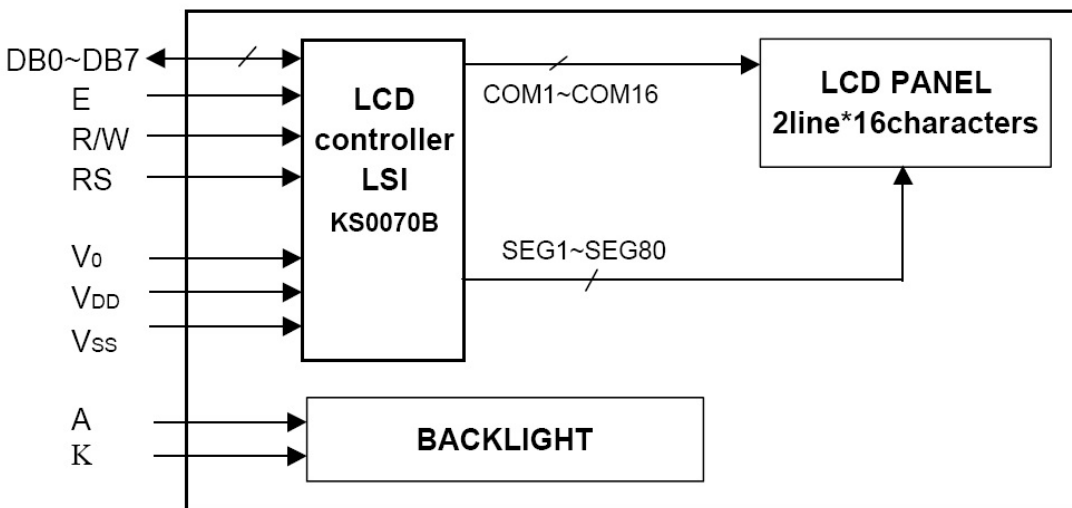
Číže nastavená hodnota  $OCR1A$  určuje dobu otvoreniu triaka. Keď je hodnota  $OCR1A$  nastavená na hodnotu 0 triak je plne otvorený a keď je nastavený na hodnotu 10000 triak je plne zatvorený.

## 5.1. RIADIACA ČASŤ

### 5.1.2. LCD displej

Prakticky všetky riadkové displeje sú riadené obvodom HD44780. V našom prípade budeme používať displej 2x8 znakov. Kvôli úspore vodičov sme vybrali komunikáciu po 4 bitovej zbernici. To znamená, že namiesto 8 dátových vodičov budeme používať 4 a zbytok uzemníme. Blokovú schému vidíme na obr. č. 5.2. Komunikácia prebieha posielaním dát alebo inštrukcií na dvakrát. Najprv posielame hornú polovicu bitu, potvrdíme kladným impulzom na E 5.1 a potom pošleme dolnú polovicu bitu a takisto vygenerujeme kladný pulz na E.

Pri inicializácii 4 bitového displeja sa najprv pošle horná polovica bitu a jeden pulz na E. Keďže displej budeme používať na zápis, môžeme komunikačný pin  $R/W$  trvalo nastaviť na zápis.



Obr. 5.2: Blokový diagram LCD displeja [29]

Zdrojový kód na obsluhu displeja bol prevzatý a upravený z [28]. Použité inštrukcie môžeme vidieť v [29] v tab.5. Z uvedených inštrukcií sú používané *LCD4\_Init*(nastavenie displeja na 4 bitovú komunikáciu), *LCD4\_Clear()*(vymazanie nápisov z displeja), *LCD4\_Home()*(posunie kurzor na začiatok prvého riadka), *LCD4\_Home2()*(posunie kurzor na začiatok druhého riadka) *LCD4\_PutS()*(Výpis reťazca char na displej). Použitie vid'. príloha č.10.7.

### 5.1.3. Sériová komunikácia

Obsluha komunikuje s riadiacim systémom pomocou počítača cez sériovú linku za pomoci USARTu. Usart je univerzálny synchrónny a asynchrónny sériový prijímač a vysielač s nasledujúcimi základnými vlastnosťami:

## 5. REALIZÁCIA RIADIACEJ JEDNOTKY

Číslo vývodu	Signál	Funkcia
1	GND	Zem
2	$U_{cc}$	Napájacie napätie (4,75 až 5,75)V
2	$U_{cc}$	Napájacie napätie (4,75 až 5,75)V
3	$U_o$	Nastavenie kontrastu displeja
4	RS8	Príkaz0, data 1)
5	$R/\bar{W}$	Čítanie (1), zápis (0) dát alebo príkazov
6	E	Vstup povolený
7	DB0	Data/príkaz(dolný bit)
8	DB1	Data/príkaz
9	DB2	Data/príkaz
10	DB3	Data/príkaz
11	DB4	Data/príkaz
12	DB5	Data/príkaz
13	DB6	Data/príkaz
14	DB7	Data/príkaz(horný bit)
15	A	Anoda k podsvieteniu LED
16	K	Katoda k podsvieteniu LED

Tabuľka 5.1: Význam vývodov displeja

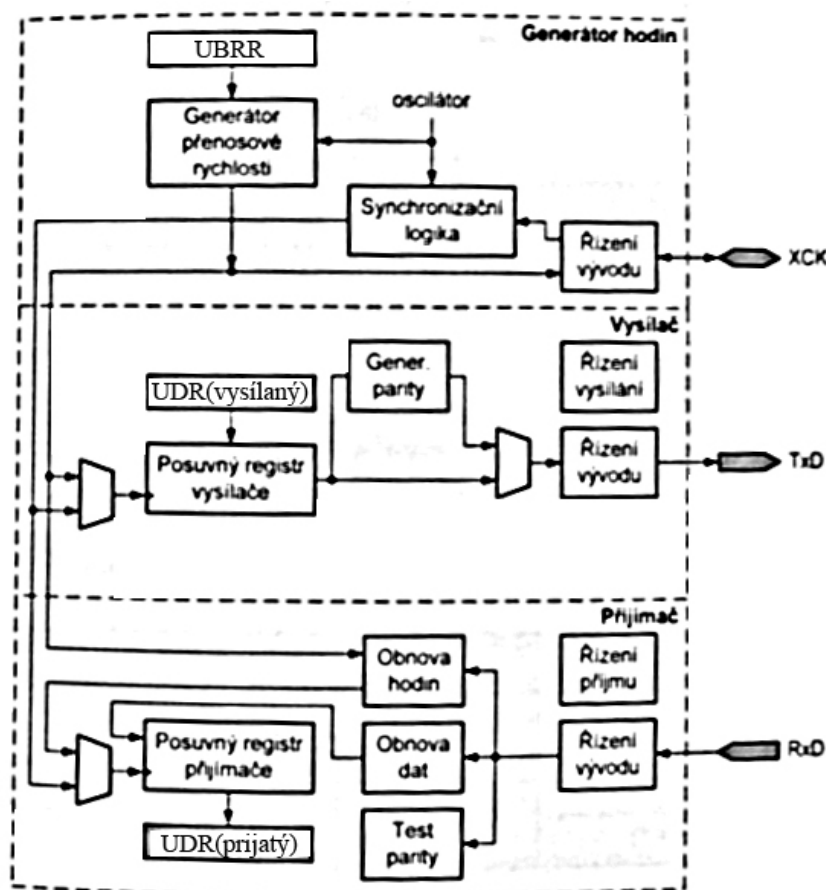
- plný duplex (môže súčasne vysielat a prijímat)
- synchronný alebo asynchronný režim
- pri synchronnom režime môže pracovať ako Master alebo ako Slave
- v asynchronnom režime je možné zdvojnásobiť prenosovú rýchlosť
- veľké rozlíšenie generátora rýchlosti(jednoduché nastavenie prenosovej rýchlosti)
- a mnoho ďalších vid'. [1].

Na obr.č. 5.3 vidíme základnú blokovú schému jednotky USART. Jeho hlavné bloky sú:

- Generátor hodín  
Obsahuje synchronizačnú logiku pre vonkajší hodinový vstup v synchronnom režime.
- Vysielač  
Obsahuje zapisovací buffer, sériový posuvný register, generátor parity a riadiacu jednotku pre obsluhu rôznych sériových rámcov.
- Prijímač  
Obsahuje jednotky pre obnovu hodín a dát pri asynchronnom príjme.Ďalej obsahuje

## 5.1. RIADIACA ČASŤ

detektor parity, riadiacu logiku, posuvný register a prijímací buffer UDR. Prijímač umožňuje detokovanie chyby rámca, chyby parity a straty znaku.



Obr. 5.3: Bloková schéma jednotky USART [1]

Hlavné funkcie USARTu sú: inicializácia, príjem dát a odosielanie dát. Pred zahájením komunikácie musí byť jednotka USART **inicializovaná**. V tomto procese je potrebné nastaviť komunikačnú rýchlosť, formát rámca a povoliť funkciu príjmu a vysielať. Na výpočet veľkosti 16 bitového registra UBRR z prenosovej rýchlosti pri normálnom asynchrónnom režime vypočítame zo vzťahu:

$$UBRR = \frac{f_o}{16 \cdot f_{BR}} - 1 \quad (5.1)$$

Pri **vysielaní** treba nastaviť vysielač nastavením bitu TXEN z registra UCSRB. Po nastavení je potrebné vložiť dáta do registra UDR, tým budú bufferované dáta posunuté do posuvného registra v okamihu, keď register bude prázdny a bude pripravený vyslať nový rámec.

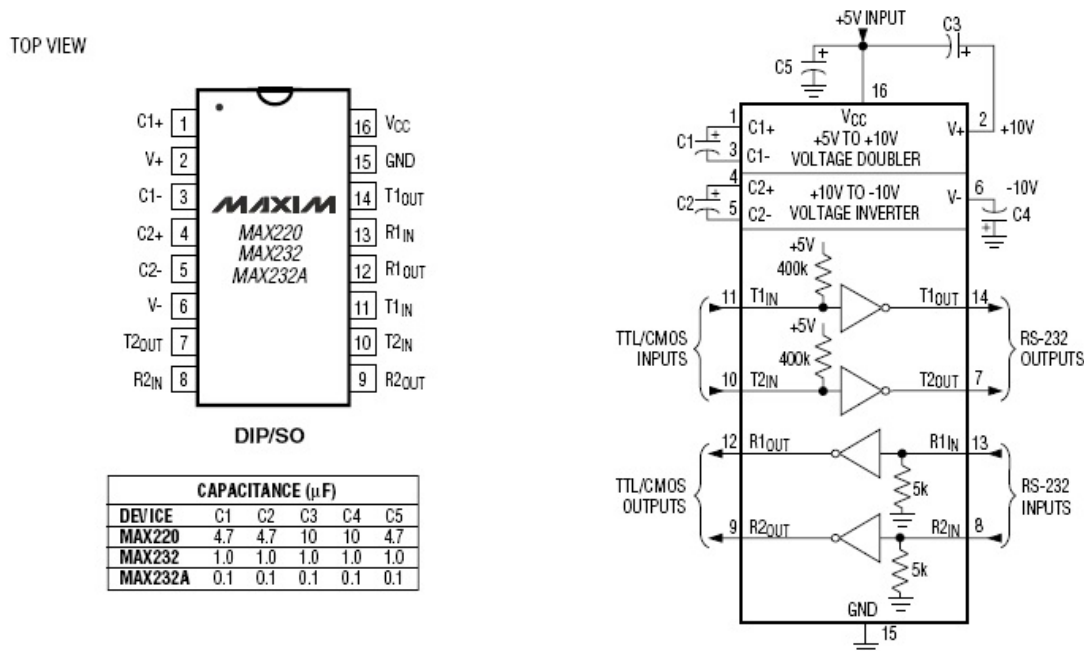
Pri **prijímaní** dát je potrebné nastaviť bit RXEN v registri UCSRB. Prijímač začne prijímať dáta, keď detekuje platný štart bit. Každý nasledujúci bit je vzorkovaný zvolenou



## 5. REALIZÁCIA RIADIACEJ JEDNOTKY

prenosovou rýchlosťou (nastavenie UBRR) a je prijímaný pokiaľ sa detekuje stop bit. Po detekcii stop bitu sa presunú prijaté sériové rámce z prijímacieho posuvného registra do prijímacieho buffera. Prijímací buffer je následne možné čítať pomocou registra UDR.

Pomocou prebratého a upraveného programu [30] sme na komunikáciu využili na príjem a odosielanie dát prerušenie. Dáta z buffera môžeme čítať pomocou funkcie *UART\_VypisBufer*. Na odosielanie dát používame funkciu *UART\_Transm* a na vymazanie dát z buffera *Flush\_UART\_Buffer*. Ďalej na komunikáciu využívame funkciu *DataInReceiveBufferOb*, ktorá nám vracia 0 keď je buffer prázdny. Použitie uvedených príkazov pri komunikácii môžeme vidieť v prílohe č. 10.6. Na komunikáciu s mikrokontrolérom potrebujeme previesť logiku z úrovne RS232 na úroveň TTL a naopak, t.j. z úrovne +15V a -15V na úroveň +5V a 0V. K tomu slúži prevodník MAX232 od firmy Maxim Integrated Circuits. Tento integrovaný obvod je napájaný zo zdroja +5V a na zvýšenie napätia využíva nábojové pumpy. Vývody pinov a základnú schému zapojenia vidíme na obr. č. 5.4.



Obr. 5.4: Vývody pinov a základná schéma zapojenia prevodníka MAX232 [34]

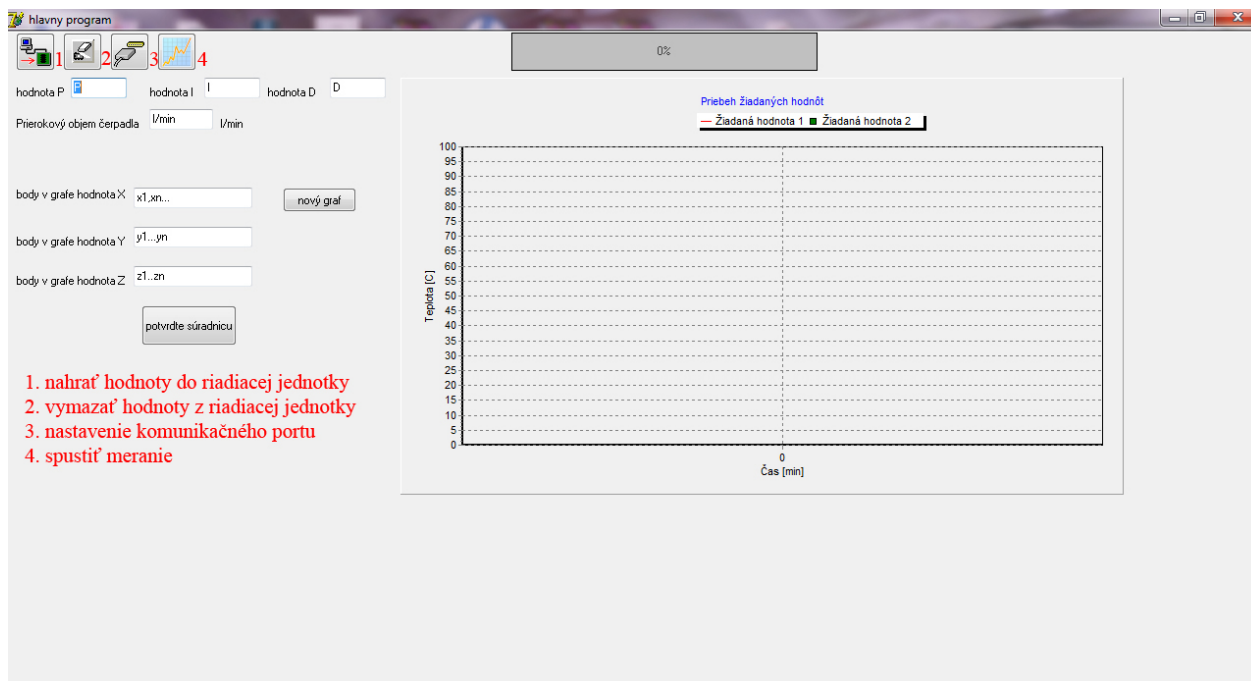
### Komunikačný softvér s PC

Na komunikáciu PC s riadiacou jednotkou slúži komunikačný softvér. Po otvorení softvéru sa nám objaví okno obr. č. 5.5, kde môžeme zadávať konštanty PSD regulátora, objemový prietok čerpadla a časovú závislosť dvoch žiadaných hodnôt, v našom prípade je sú to

## 5.1. RIADIACA ČASŤ

veličiny pre rmutovací postup, a to je teplota a objem. Program súčasne vykresľuje aj priebehy zadaných hodnôt. Pri zadávaní hodnôt je potrebné potvrdiť hodnoty veličín, pre každý časový úsek. K tomu slúži tlačítko "Potvrdiť súradnicu". Pre prekopírovanie týchto hodnôt je potrebné vybrať komunikačný port (tlač.3.) a nahráť dáta do riadiacej jednotky (tlač.1.). Po stlačení tlačítka (tlač.1.) sa nám na displeji riadiacej jednotky objaví výpis "Počet prenesených hodnôt" a po úspešnom uložení do pamäte EEPROM sa objaví výpis "Uspešne uloženie dat".

Stlačením (tlač.2.) vymažeme hodnoty žiadaných veličín z riadiacej jednotky (vymaže pamäť EEPROM). Stlačením tlačítka (tlač.4.) a následným stlačným tlačítka "Start" po otvorení meracieho okna môžeme spustiť meranie, kde môžeme vidieť priebehy žiadanej a nameranej veličiny a priebehy riadiaceho uhla v závislosti na čase.



Obr. 5.5: Základný popis komunikačného softvéru

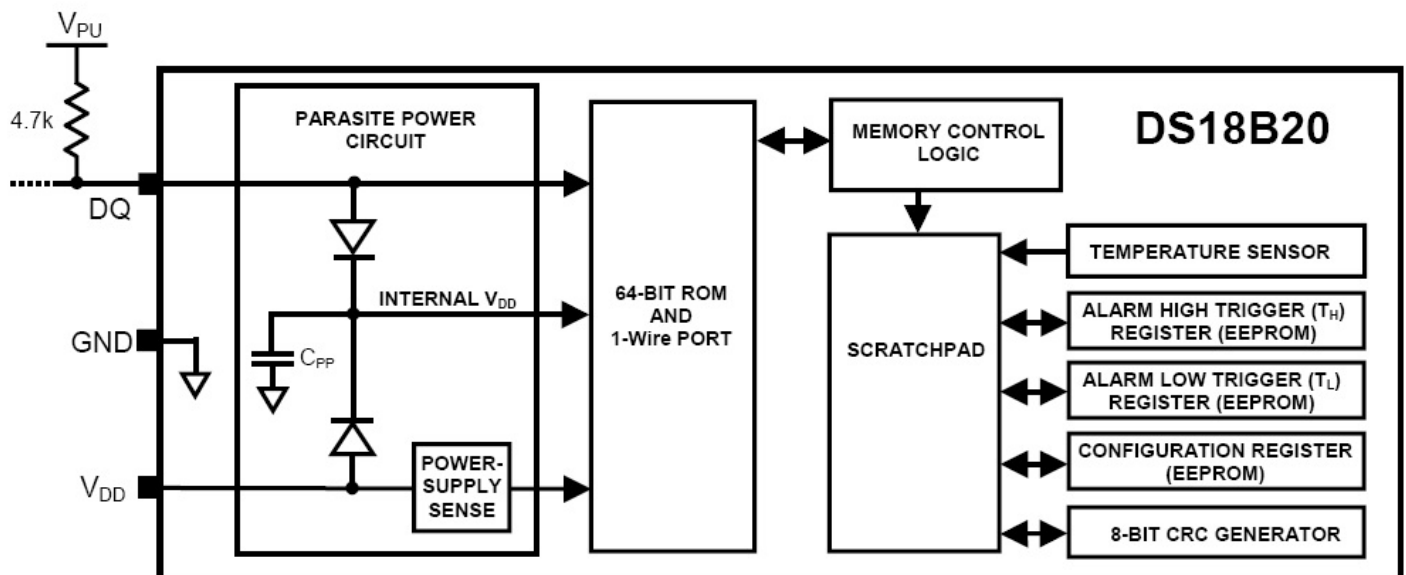
### 5.1.4. Čidlo DS18B20

Čidlo DS18B20 na komunikáciu využíva 1Wire zbernicu, čo na komunikáciu vyžaduje jeden datový vodič. Teplotný rozsah čidla je  $(-55 \text{ až } 125)^{\circ}\text{C}$  s presnosťou  $\pm 0,5^{\circ}\text{C}$  v rozsahu  $(-10 \text{ až } 85)^{\circ}\text{C}$  a v krajných medziach je presnosť prevodu  $\pm 2^{\circ}\text{C}$ . Integráciou potrebných obvodov do jedného púzdra boli minimalizované vývody na 3, t.j. VDD(napájanie), GND(zem), DQ(dátový vodič).

## 5. REALIZÁCIA RIADIACEJ JEDNOTKY

Na obr.5.6 vidíme blokovú schému inteligentného teplotného čidla DS18B20. Jeho hlavné časti sú :

- 64-bitový ROM
- scratchpad
- snímač teploty a prevodník na digitálne číslo
- konfiguračný register
- registre pre nastavenie alarmu



Obr. 5.6: Blokový diagram teplotného čidla DS18B20 [25]

### Popis a nastavenie komunikácie

Komunikácia čidla DS18B20 cez 1-wire zbernicu prebieha nasledovne:

- Reset
- ROM príkazy
- Pamäťové príkazy
- Vysielanie a príjem dát

## 5.1. RIADIACA ČASŤ

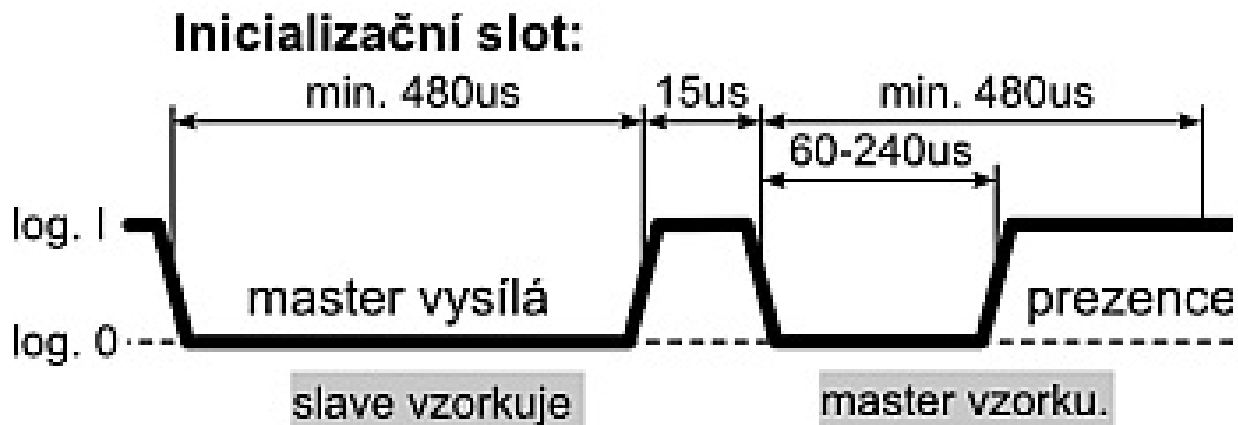
- Reset

Komunikáciu zahajujeme reset pulzom. Po ohlásení zariadenie začne prijímať a vysielat dáta, ktoré sú vysielané v "time slotoch". Existujú 4 druhy time slotov:

- zápis logickej 1
- zápis logickej 0
- čítanie logickej 1
- čítanie logickej 0

### Reset

Na obr.5.7 vidíme priebeh resetovacieho pulzu po 1wire zbernici. Na začiatku je pomocou mikroprocesora nastavená logická 0 po dobu  $480\ \mu\text{s}$ . Potom nasleduje uvoľnenie zbernice, hodnota sa pomocou pull-up rezistora vráti na hodnotu logickej 1 a čaká na prezence signál(logická 0) od slave, ktorá by mala mať dĺžku (15 až  $60\ \mu\text{s}$ ). Tento postup slúži na kontrolu správnej pripojiteľnosti čidla a je potrebné ho urobiť vždy pred začiatkom každého merania. Po resetovacom pulze a úspešnej odozve môžeme začať vysielat a prijímať



Obr. 5.7: Reset 1-wire zbernice [26]

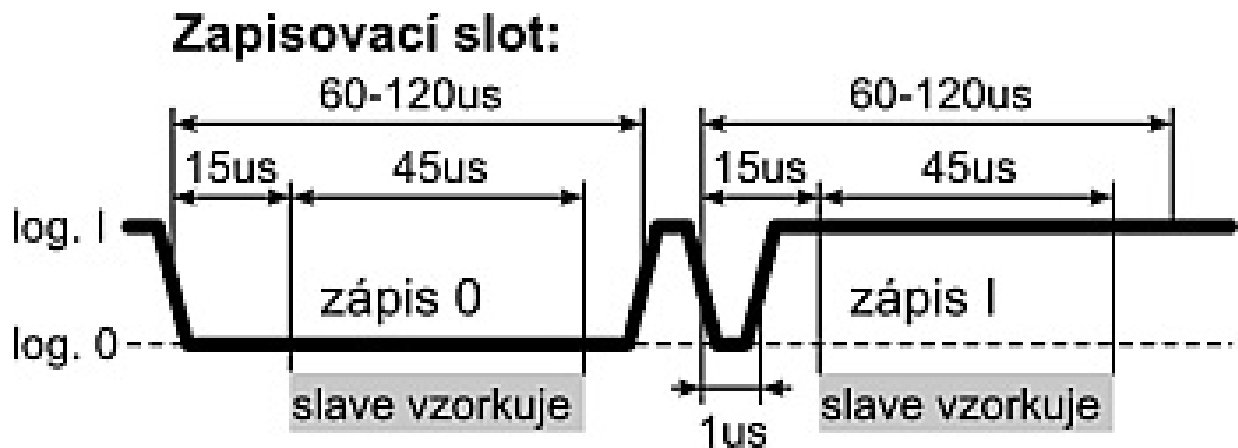
dáta.

### Zapisovací slot

Zapisovací slot môžeme rozdeliť na dva typy. Na zápis logickej 1 a na zápis logickej 0. Na

obr.5.8 vidíme priebeh pulzu zapisovacieho slotu po 1wire zbernici.

- Pre zápis logickej 0 je potrebné stiahnuť zbernicu na úroveň logickej 0 a držať ju na nej po dobu  $(60 \text{ až } 120) \mu s$ .
- Pre zápis logickej 1 je potrebné stiahnuť zbernicu na úroveň logickej 0 a držať ju na nej po dobu  $(1 \text{ až } 15) \mu s$ . Po uvoľnení sa zbernica zdvihne pomocou pull-up rezistora na úroveň log.1.



Obr. 5.8: Vysielanie dat po 1-wire zbernici [26]

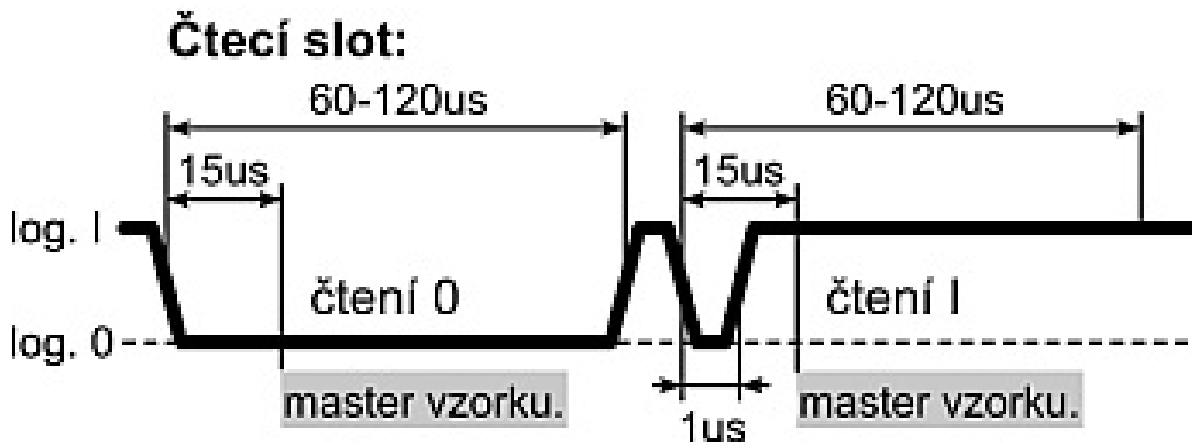
### Čítací slot

Na obr.5.9 vidíme priebeh pulzu čítacieho slotu po 1wire zbernici.

Čítanie logickej hodnoty je zahájené stiahnutím zbernice na úroveň logickej 0 po dobu  $1 \mu s$ . Po tomto zahájení môže čidlo:

- vyslať logickú 1 tým, že po uplynutí  $15 \mu s$  ponechá zbernicu v log.1) alebo
- vyslať logickú 0 tým, že po uplynutí  $15 \mu s$  stiahne zbernicu na úroveň logickej 0 a ponechú ju tam po dobu  $45 \mu s$ .

Po čítaní alebo zápise musíme nechať slot pre regeneráciu v kľude po dobu  $1 \mu s$ .



Obr. 5.9: Příjem dat po 1-wire zbernici [26]

### ROM příkazy

Po resete a zaznamenání prezencie pulzu můžeme vyslat jeden z nasledujících ROM povelov:

- Čítaj ROM [0x33]  
Tento príkaz slúži na zistenie 8-bitového rodinného kódu, jedinečného 48-bitového sériového čísla a 8-bitového CRC.
- Porovnaj ROM [0x55]  
Tento príkaz dovolí mikrokontroléru osloviť špecifický obvod na zbernici.
- Preskoč ROM [0xCC]  
Tento príkaz nám dovoľuje sprístupniť pamäťové funkcie bez znalosti ROM kódu, avšak iba u oslovených zariadeniach.
- Vyhľadaj ROM [0xF0]

Tento príkaz dovolí mikrokontroléru používať proces vedúci k zisteniu 64-bitových ROM kódu všetkých zariadení, pripojených na zbernicu.

- Hľadaj alarm [0xEC]  
Tento príkaz je funkčný, ak sa hodnota nachádza nad maximálnou alebo pod minimálnou hodnotou alarmu.

**Pamäťové príkazy**

U teplotného čidla DS18B20 môžeme použiť nasledovné pamäťové príkazy:

- Zápis do Scratchpadu [0x4E]

Tento príkaz slúži na zápis informácií do scratchpadu. Začína sa s najnižším bajtom (nižší bajt teplotného registra) a postupne môžeme pokračovať ku konfiguračnému registru.

- Čítaj Scratchpad [0xBE]

Tento príkaz slúži na prečítanie celého obsahu scratchpadu, od prvého až po deviaty bajt.

- Kopíruj Scratchpad [0x48]

Príkaz skopíruje obsah scratchpadu do pamäte EEPROM čidla DS18B20.

- Preveď teplotu [0x44]

Tento príkaz slúži na štartovanie teplotnej konverzie a výsledok uloží do scratchpadu.

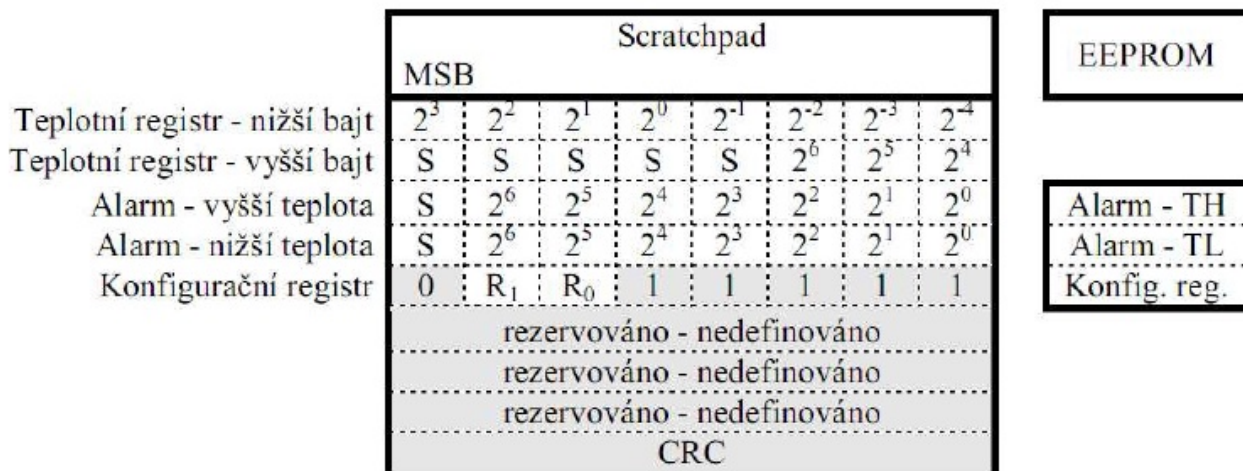
- Čítaj z EEPROM [0xB8]

Tento príkaz presúva hodnoty z EEPROM pamäte do scratchpadu.

Ako je vidieť na obr.5.10 pamäť obvodu DS18B20 sa skladá z pracovných registrov RAM (scratchpad) a z pamäte EEPROM. Datá je možné ukladať do scratchpadu pomocou príkazu [0x4E]. Pre kontrolu môžeme použiť príkaz na čítanie dát zo scratchpadu [0x4E] a po overení správnosti zápisu môžeme prekopírovať obsah scratchpadu do pamäte EEPROM pomocou príkazu [0x48]. Pre naše využitie sú dôležité prvé dva registre zo scratchpadu, ktoré obsahujú nižší a vyšší bajt teplotného registra, prípadne piaty (konfiguračný) register, ktorý slúži na nastavenie rozlíšenia teploty. Tretí a štvrtý register nám umožňuje nastavenie hornej a dolnej hranice alarmu. Pri prekročení týchto hraníc sa v obvode nastaví príznak alarmu.

Čidlo DS18B20 využíva priamy prevod teploty na digitálny signál. Po príkaze **Preveď teplotu** [0x44] čidlo DS18B20 uloží binárne dáta do scratchpadu v 16 bitovom formáte, viď obr.5.2 Prvých 5 bitov nám udáva znamienko, ďalších 7 nám udáva celočíselnú časť nameranej hodnoty (s rozsahom -55 až 125) a zvyšné 4 udávajú hodnotu teploty za desatinnou čiarkou s maximálnym rozlíšením  $0,0625^{\text{circ}}\text{C}$ . Znižovaním rozlíšenia zvyšujeme

## 5.2. VÝKONOVÁ ČASŤ



Obr. 5.10: Pamäť obvodu DS18B20 [26]

rýchlosť prevodu. Tieto informácie môžeme získať pomocou príkazu **Čítaj Scratchpad** [0xBE]. Namerané dáta budú vyslané pomocou 1-wire zbernice, kde sa ako prvý vyšle najnižší bajt LSB a následne sa vyšle najvyšší bajt MSB. Pre podrobnejšie informácie viď. [25] ,prípadne [26].

Teplotný register			
Teplotný register-vyšší byt	Teplotný register-nížší byt	teplota[HEX]	teplota[°C]
0000 0111	1101 0000	07 D0	+125,000
0000 0101	0101 0000	05 50	+85,000
0000 0001	1001 0001	01 91	+25,0625
0000 0000	1010 0010	00 A2	+10,1250
0000 0000	0000 1000	00 08	+0,50000
0000 0000	0000 0000	00 00	0
1111 1111	1111 1000	FF F8	-0,50000
1111 1111	0101 1110	FF 5E	-10,1250
1111 1110	0110 1111	FF 6F	-25,0625
1111 1100	1001 0000	FC 90	-55,0000

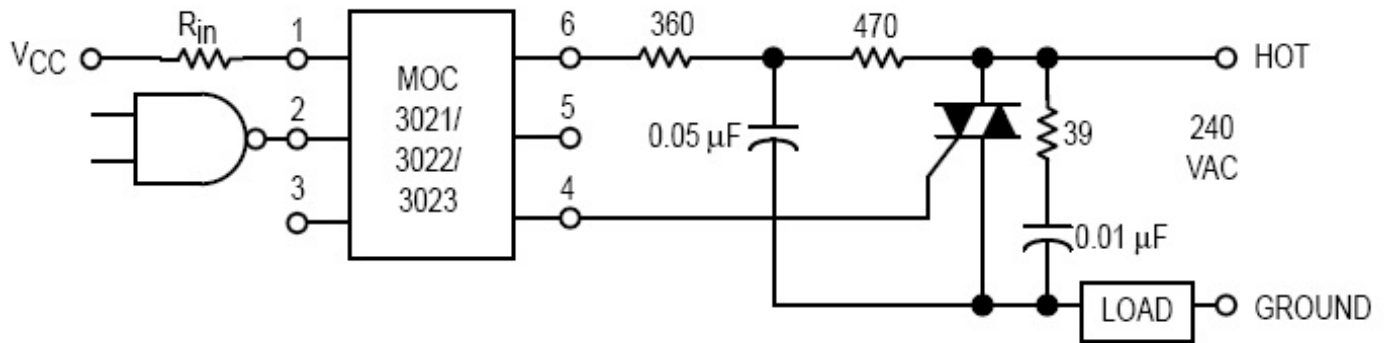
Tabuľka 5.2: Teplotný register čidla DS18B20 [26]

Softvérovú realizáciu teplotného čidla DS18B20 môžeme vidieť v 10.3.

## 5.2. Výkonová časť

Riadiaca časť riadiacej jednotky je galvanicky oddelená od výkonovej časti pomocou optotriakov typu *MOC3022*. Výkonová časť je napájaná zo sieťového napätia, t.j. striedavým napätím 230 V a slúži na spínanie výkonových súčiastok. V našom prípade výkonovú časť





Obr. 5.11: odporúčané zapojenie optotriaka s triakom [31]

tvorí jeden triak a dve relé. Triak je riadený pomocou PWM a môžeme s ním regulovať záťaž s maximálnym výkonom 3,6 kW [32]. Výrobcom odporúčané zapojenie optočlena s triakom je vidieť na obr. č. 5.11. Relé slúži na zapínanie a vypínanie spotrebiča s maximálnym prúdovým zaťažením 10 A pri menovitom napätí 230V [33].

Kompletné zapojenie výkonovej časti je vidieť v prílohe č. 10.1 a 10.2.

## 6. Prezentácia výsledkov

### 6.1. Realizácia plošného spoja

#### 6.1.1. Výkonová časť

Na obr. č. 6.1, prípadne v prílohe č.10.1 vidíme zapojenie výkonovej časti, ktorá sa skladá



Obr. 6.1: obrázok plošného spoja výkonovej časti

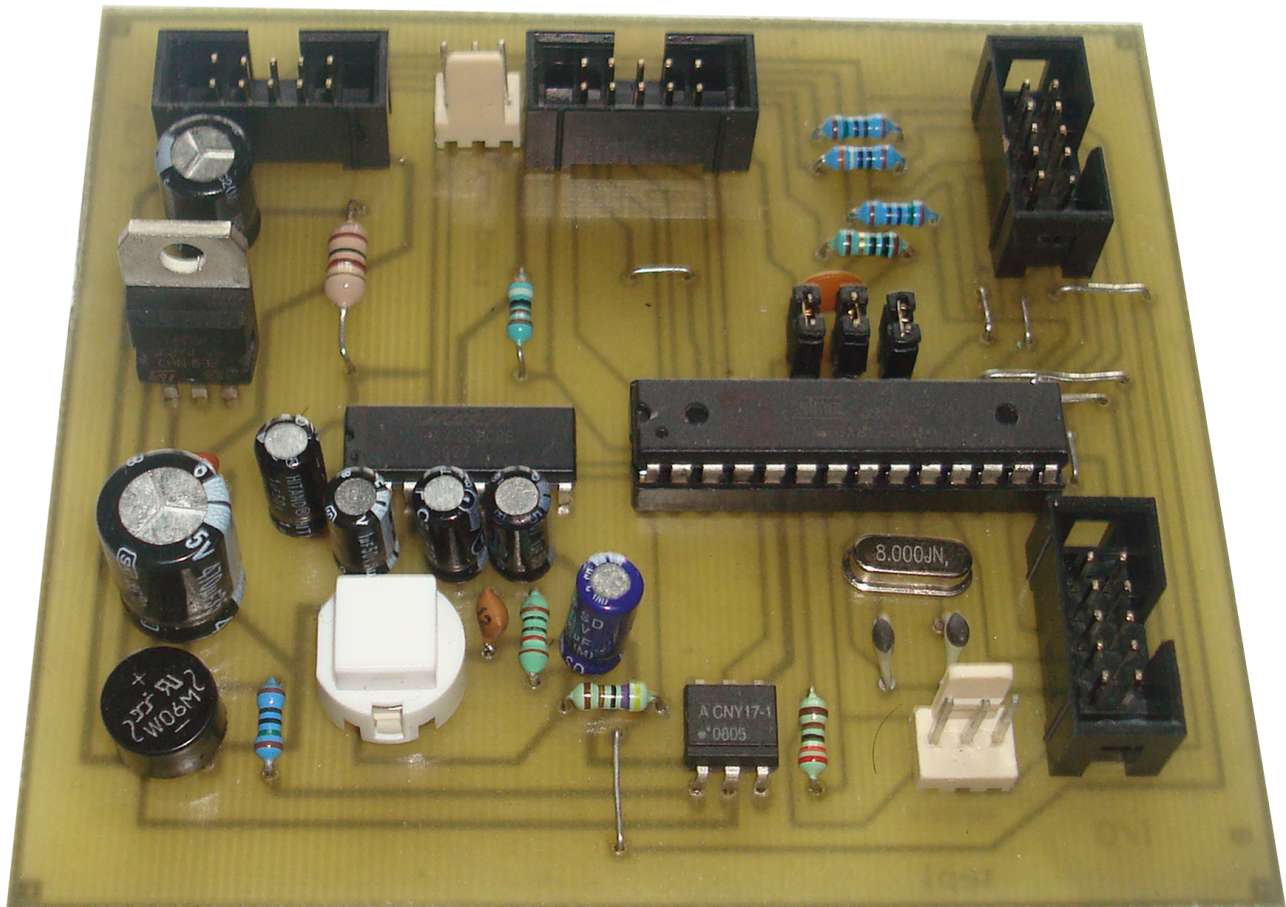
z nasledovných hlavných častí:

- Triak,  
využíva uhlové riadenie pomocou PWM na reguláciu výstupného výkonu.
- Optotriak MOC3022  
Slúži na galvanické oddelenie riadiacej a výkonovej časti.

- Relé,  
zapína a vypína výstup. Cyklus zapínania a vypínania je riadený pomocou programu.
- Transformátor,  
slúži na pretransformovanie napätia na 6V, pre napájanie riadiacej časti.

### 6.1.2. Riadiaca časť

Na obr. č. 6.2, prípadne v prílohe č.10.3 vidíme zapojenie riadiacej časti, ktorá sa skladá



Obr. 6.2: obrázok plošného spoja riadiacej časti

z nasledovných hlavných častí:

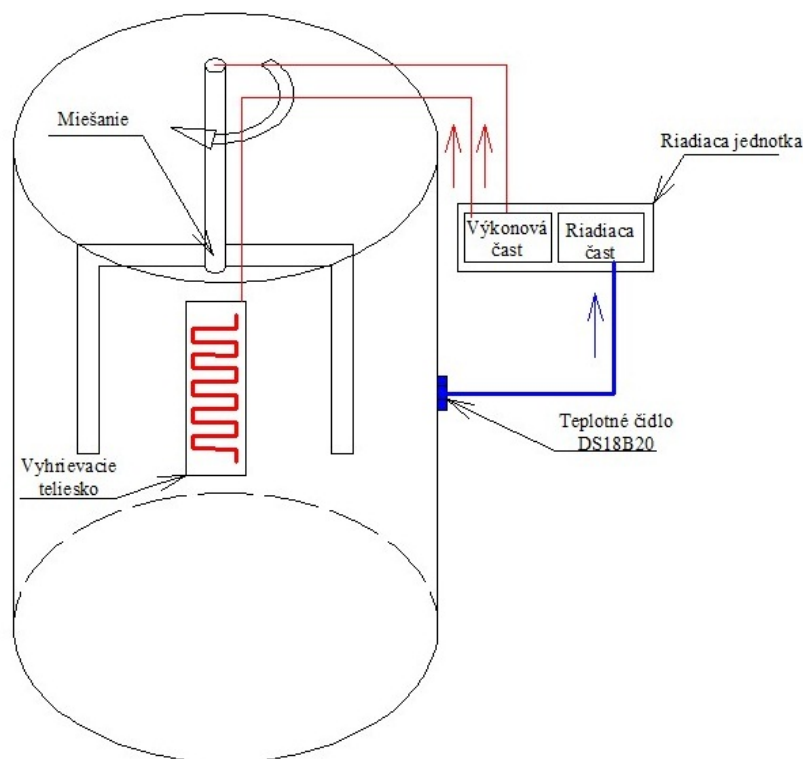
- Mikrokontrolér Atmel Atmega 8,  
programovateľný jednočip, ktorý je spolu so svojimi perifériami hlavnou časťou riadiacej jednotky.

## 6.2. NAMERANÉ HODNOTY

- Diódový mostík ,stabilizátor 7805  
spolu s kondenzátormi slúžia na prevod striedavého napätia na jednosmerné a na jeho stabilizáciu na +5V, pre zapojenie mikrokontroléra.
- MAX232,  
integrovateľný obvod slúžiaci na prevod logiky z TTL úrovne na úroveň RS232 a naopak. Používa sa pri sériovej komunikácii.
- Optočlen CNY 17,  
služí na galvanické oddelenie a na spínanie externého prerušenia pri prechode sieťového napätia nulou.

## 6.2. Namerané hodnoty

Funkčnosť riadiacej jednotky sme vyskúšali na sústave, ktorého schéma je uvedená na obr.č.6.3.

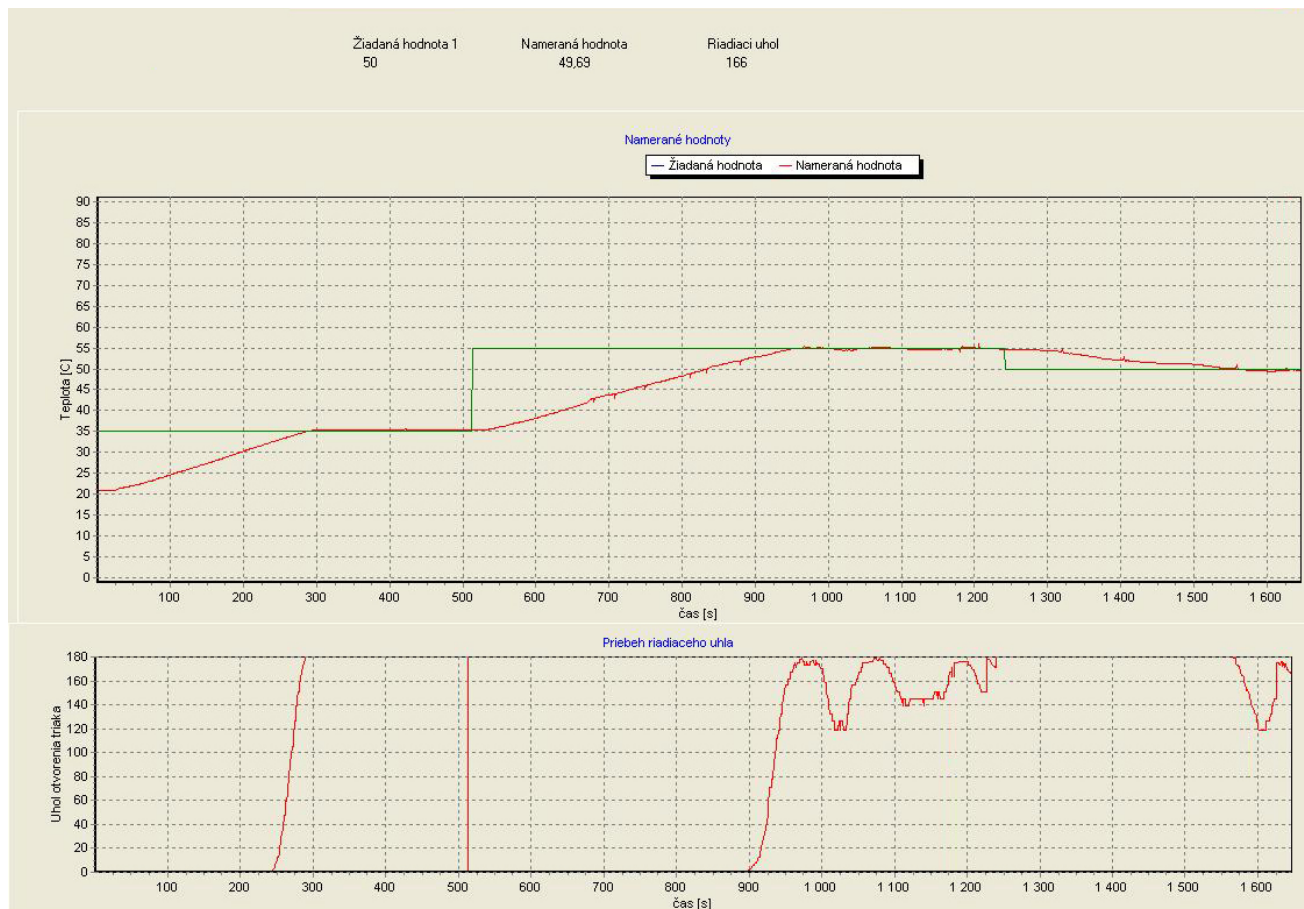


Obr. 6.3: Schéma merania

Sústava sa skladá z plechovej nádoby s objemom 2l. Na ohrievanie sme použili ponorný varič s výkonom 500W. Namerané hodnoty sú zobrazené na obr. č. 6.4. Počas celého merania bolo zapnuté miešanie, aby teplota vody bola rovnaká v celom objeme. Na obrázku



6.5 vidíme detailný pohľad z obr. č. 6.4. Obrázok nám znázorňuje presnosť regulácie pri teplote  $55\text{ }^{\circ}\text{C}$ .



Obr. 6.4: Pribeh merania teploty a uhla riadenia v závislosti na čase

Na uvedených obrázkoch vidíme priebehy žiadanej teploty (zelená), nameranej teploty (červená) v  $^{\circ}\text{C}$  a priebeh riadiaceho uhla v  $^{\circ}$  v závislosti na čase v  $[\text{s}]$ . Pri zadávaní žiadanej hodnoty sa zadáva výdrž na žiadanej hodnote, t.j. časový úsek výdrže teploty po dosiahnutí žiadanej hodnoty. V našom prípade bolo zadané:

- 3min výdrž na  $35^{\circ}\text{C}$
- 5min výdrž na  $55^{\circ}\text{C}$
- 2min výdrž na  $50^{\circ}\text{C}$



Obr. 6.5: Detailný pohľad udržovania teploty z obr. č. 6.4

## 6.2. NAMERANÉ HODNOTY

Nepresnosť merania môže byť spôsobená nepresnosťou čidla  $\pm 0,5[^\circ C]$ . Presnosť regulácie je závislá od rozdielu medzi teplotou okolia a žiadanou teplotou. Pri väčšom rozdieli dôjde k rýchlejšej disipácii tepla a tým k rýchlejšiemu ustáleniu sústavy.

## 7. Záver

Cieľom tejto diplomovej práce bol návrh a realizácia univerzálnej riadiacej jednotky pre riadenie jednoduchých technologických procesov v potravinárskom priemysle. Riadiaca jednotka je navrhnutá na výrobu piva, kde potrebujeme udržiavať teplotu s toleranciou  $\pm 1\text{ }^{\circ}\text{C}$ . Pri meraní sme dosiahli prekmit s maximálnou hodnotou  $0,6\text{ }^{\circ}\text{C}$ . Po ustálení prechodového deja je priemerná tolerancia  $\pm 0,3\text{ }^{\circ}\text{C}$  a je závislá od rozdielu žiadanej a okolitej teploty. V prípade, keď teplota okolia dosiahne vyššiu hodnotu ako je požadovaná, navrhnutá sústava je neregulovateľná. V takom prípade by bolo potrebné použiť chladenie, prípadne chladenie s reguláciou. Riadiaca jednotka je navyše vybavená konektorom s možnosťou pripojenia ďalších periférií ako napr. snímače, tlačítka (a iné vstupy alebo výstupy...). Riadiacu jednotku by okrem spomínanej regulácie chladenia bolo možné vylepšiť aj automatickým nastavením koeficientov regulátora PID, prípadne na komunikáciu použiť bezdrôtovú sieť WIFI, s ktorou už niektoré typy mikrokontrolérov dokážu komunikovať. Uvedené vylepšenia neboli súčasťou zadania diplomovej práce, boli iba návrhom pre ďalší krok pri zjednodušení riadenia so zvýšením presnosti.

## 8. Literatura a odkazy

- [1] DAVID MATOUŠEK: *Práce s mikrokontroléry Atmel AVR*, BEN-technická literatura Praha, 2006
- [2] VLADIMÍR VÁŇA: *Mikrokontroléry Atmel AVR Popis procesoru a inštrukční soubor*, BEN-technická literatura Praha, 2003
- [3] VLADIMÍR VÁŇA: *Mikrokontroléry Atmel AVR programování v jazyce C*, BEN-technická literatura Praha, 2003
- [4] KOSAŘ, K; PROCHÁDZKA, S.: *Technologie výroby sladu a piva.*, Výzkumný ústav pivovarský a sladařský a.s. Praha, , 2000. ISBN 80-902658-6-3.
- [5] IVAN ŠVARC: *ZÁKLADY AUTOMATIZACE*, Skriptum VUT Brno, 2002
- [6] JIŘÍ SKALICKÝ: *Teorie řízení*, Skriptum VUT Brno, 2002
- [7] *Výroba piva* [online].1999.[cit.2009-11-15].Příprava a výroba mladiny . Dostupné z WWW: <http://www.pivo.sk/pivo/vyroba/rozdelen/mladina.htm>
- [8] *Elitronic* [online].2007.[cit.2009-11-16].DistEl TER 640 . Dostupné z WWW: [http://www.elitronic.cz/vyr\\_techcelky.html](http://www.elitronic.cz/vyr_techcelky.html)
- [9] *Triangle Research International* [online].2009.[cit.2009-11-17].T100MD888+ . Dostupné z WWW: <http://www.tri-plc.com/t100md888.htm>
- [10] *MRK - meracia a regulačná technika* [online].[cit.2009-11-17].Programovatelný regulátor RR. Dostupné z WWW: <http://www.mrk.eu/rr.htm>
- [11] *SEA-Společnost pro elektronické aplikace* [online].2006.[cit.2009-11-18].FBS-10MA . Dostupné z WWW: [http://www.seapraha.cz/index.php?action=show\\_cat&cPath=114\\_137\\_139\\_171](http://www.seapraha.cz/index.php?action=show_cat&cPath=114_137_139_171)
- [12] *PLC-Teco* [online].[cit. 2009-11-18].PLC TECOMAT TC 400. Dostupné z WWW:[http://www.plc-control.net/plc.htm?id\\_znacky=1](http://www.plc-control.net/plc.htm?id_znacky=1)
- [13] *Tecomat* [online].2005.[cit.2009-11-18].TECOMAT TC400 - kompaktní PLC. Dostupné z WWW: <http://www.tecomat.cz/index.php?lang=cs&m1id=1&m2id=4&m3id=11&mid=10>



- [14] *Axima* [online].2006-03 .[cit.2009-11-18].Ceník OPLC UNITRONICS.Dostupné z WWW: [http://www.axima.cz/web/eu/pdf/automatizace\\_CZ\\_0\\_UNITRONICS\\_3.pdf](http://www.axima.cz/web/eu/pdf/automatizace_CZ_0_UNITRONICS_3.pdf)
- [15] ANDRT, Martin; FALC, Peret.Nové automaty Jazz překračují všechna pravidla. *Automa* [online].2006, 03.[cit.2009-11-19]. Dostupné z WWW: [http://www.odbornecasopisy.cz/index.php?id\\_document=31034](http://www.odbornecasopisy.cz/index.php?id_document=31034)
- [16] *V - system elektro* [online].2006.11.[cit.2010-01-24].Triakový regulátor EFM.Dostupné z WWW: [http://www.v-system.cz/Files/861\\_EFM\\_navod\\_CZ.pdf](http://www.v-system.cz/Files/861_EFM_navod_CZ.pdf)
- [17] *V-system elektro* [online].[cit.2010-01-24].EFM-9161 - Regulátor výkonu.Dostupné z WWW: <http://eshop.v-system.cz/DetailPage.asp?DPG=22080>
- [18] *OES* [online].1998.[cit.2010-01-24].High Performance Solenoid Driver, Pick and Hold Module. Dostupné z WWW: <http://www.motioncontrolplanet.com/pickholdmodule.htm>
- [19] *Moeller*[online].2009.[cit.2010-01-24].PS4-201 přizpůsobivá jednotka.Dostupné z WWW: <http://www.zatah.moeller.cz/priruckazapojeni/control035.html>
- [20] *Saze* [online].2009.[cit.2010-01-24].Řídící systém OMRON CPM1A-10CDR-A-V1.Dostupné z WWW: <http://www.saze.cz/ridici-system-omron-cpm1a10cdrav1/automatizace/531.html>
- [21] *atpjournal.sk*. [online].2008-5 [cit.2010-01-24].CP1L kompaktný PLC automat Omron. Dostupné z WWW: [http://www.atpjournal.sk/casopisy/atp\\_08/pdf/atp-2008-05-26.pdf](http://www.atpjournal.sk/casopisy/atp_08/pdf/atp-2008-05-26.pdf)
- [22] *SCT-Telemetry*[online].2009.[cit.2010-01-25].SCT-PLC 5BINOUT.Dostupné z WWW: [http://www.sct-telemetry.eu/index.php?option=com\\_content&view=article&id=11%3AAsc&Itemid=45&lang=sk](http://www.sct-telemetry.eu/index.php?option=com_content&view=article&id=11%3AAsc&Itemid=45&lang=sk)
- [23] Wescott , Tim. *Embedded.com* [online].[cit.2009-11-17]. PID Without a PhD . Dostupné z WWW: <http://www.embedded.com/2000/0010/0010feat3.htm>
- [24] Smutný, Lubomír. *AUTOMA* [online]. 2007.[cit. 2009-11-17]. Snímače teploty současný stav a směry vývoje. Dostupné z WWW: [http://www.odbornecasopisy.cz/index.php?id\\_document=34041](http://www.odbornecasopisy.cz/index.php?id_document=34041)

- [25] *MAXIM DS18B20 - Datasheet* [online]. Dallas Semiconductor, 2008.[cit. 2009-11-17]. Dostupný z <http://datasheets.maxim-ic.com/en/ds/DS18B20.pdf>
- [26] *HW.CZ* [online]. 2004-10-17.[cit.2009-11-17]. Sběrnice 1-Wire. Dostupné z WWW: <http://hw.cz/rozhrani/art1215-sbernice-1-wire.html>
- [27] ŠVÍKA, Jan. *HappyDigi* [online]. 2009-11-30.[cit.2010-01-25]. Popis obvodu DS18B20. Dostupné z WWW: [http://happydigi.ic.cz/clanky/DS18B20\\_popis.html](http://happydigi.ic.cz/clanky/DS18B20_popis.html)
- [28] WICTORSSON, Lars. *PUDN Programmers* [online].2004-12-09.[cit.2009-12-17]. LCD4BITS. Dostupné z WWW: [http://read.pudn.com/downloads42/sourcecode/embed/144385/LCD4BIT.C\\_.htm](http://read.pudn.com/downloads42/sourcecode/embed/144385/LCD4BIT.C_.htm)
- [29] *LCD MODULE - Datasheet* [online]. Display ElektroniK GmbH, 2001.[cit.2010-01-20]. Dostupný z WWW:<http://www.maplin.co.uk/media/pdfs/N27AZ.pdf>
- [30] *USART - Zdrojový kód* [online]. 2002.[cit. 2010-01-26]. Dostupný z WWW: <http://code.google.com/p/reflowoven/source/browse/trunk/USART2.c?spec=svn9&r=9>
- [31] *Motorola - MOC3022 Datasheet* [online]. STMicroelectronics, 1995.[cit.2010-01-20]. Dostupný z WWW: [http://www.ic-on-line.cn/IOL/datasheet/moc3021\\_310597.pdf](http://www.ic-on-line.cn/IOL/datasheet/moc3021_310597.pdf)
- [32] *ST - BTA16A Datasheet* [online]. STMicroelectronics, 2010.[cit.2010-01-21]. Dostupný z WWW: <http://www.st.com/stonline/books/pdf/docs/7471.pdf>
- [33] *Finder-40series Datasheet* [online]. Finder, 2001.[cit.2009-11-21]. Dostupný z WWW: <http://www.finder.cz/comuni/pdf/S40EN.pdf>
- [34] *MAXIM - MAX232 Datasheet* [online]. Maxim, 2000.[cit.2009-11-28]. Dostupný z WWW: <http://www.finder.cz/comuni/pdf/S40EN.pdf>
- [35] *Seebeckov jav. In Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2010.[cit.2010-2-28]. Dostupný z WWW: [http://sk.wikipedia.org/wiki/Seebeckov\\_jav](http://sk.wikipedia.org/wiki/Seebeckov_jav)
- [36] FERDA, JOSEF.: *Automatické řízení postřikové dávky zemědělského stroje.*, Fakulta strojního inženýrství,Brno, 2006. 80 s. Diplomová práce. Vysoké učení technické.,

- [37] *Potravinářství perspektivní trh pro automatizaci. AUTOMA* [online]. 12.2007 [cit. 2010-3-24]. Dostupný z WWW: [http://www.odbornecasopisy.cz/index.php?id\\_document=36378](http://www.odbornecasopisy.cz/index.php?id_document=36378) .
- [38] Karas, Ondřej. *Kvetakov.net* [online]. 2008 [cit. 2009-11-25]. Články - AVR. Dostupné z WWW: <http://www.kvetakov.net/clanky/avr/>. ISSN 1803-1021.
- [39] *Mikrokontrolér. In Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, , last modified on 25. február 2010 [cit. 2010-05-25]. Dostupné z WWW: <http://sk.wikipedia.org/wiki/Mikrokontrol%C3%A9r>.
- [40] DŘÍNEK, Milan. *RISC versus CISC architektura. Avr.hw.cz* [online]. 12.01.2000, [cit. 2010-05-25]. Dostupný z WWW: [http://avr.hw.cz/architektura/risc\\_cisc.html](http://avr.hw.cz/architektura/risc_cisc.html)
- [41] *Atmega8 Datasheet* [online]. 12.2003, [cit. 2009-11-11]. Dostupný z WWW: [http://www.gme.sk/\\_dokumentace/dokumenty/432/432-201/dsh.432-201.1.pdf](http://www.gme.sk/_dokumentace/dokumenty/432/432-201/dsh.432-201.1.pdf)
- [42] HANKOVEC, David. *Dhservis* [online]. 2002 [cit. 2009-11-15]. Univerální stmívač. Dostupné z WWW: <http://www.dhservis.cz/dalsi/stmivac.htm>
- [43] Karas, Ondřej. *AVR - obecně* [online]. 2008-6-17 [cit. 2009-1-25]. Články - AVR. Dostupné z WWW: <http://programujte.com/?rubrika=70-elektrotechnika&sekce=179-avr>. ISSN 1801-1586.

## 9. Zoznam príloh

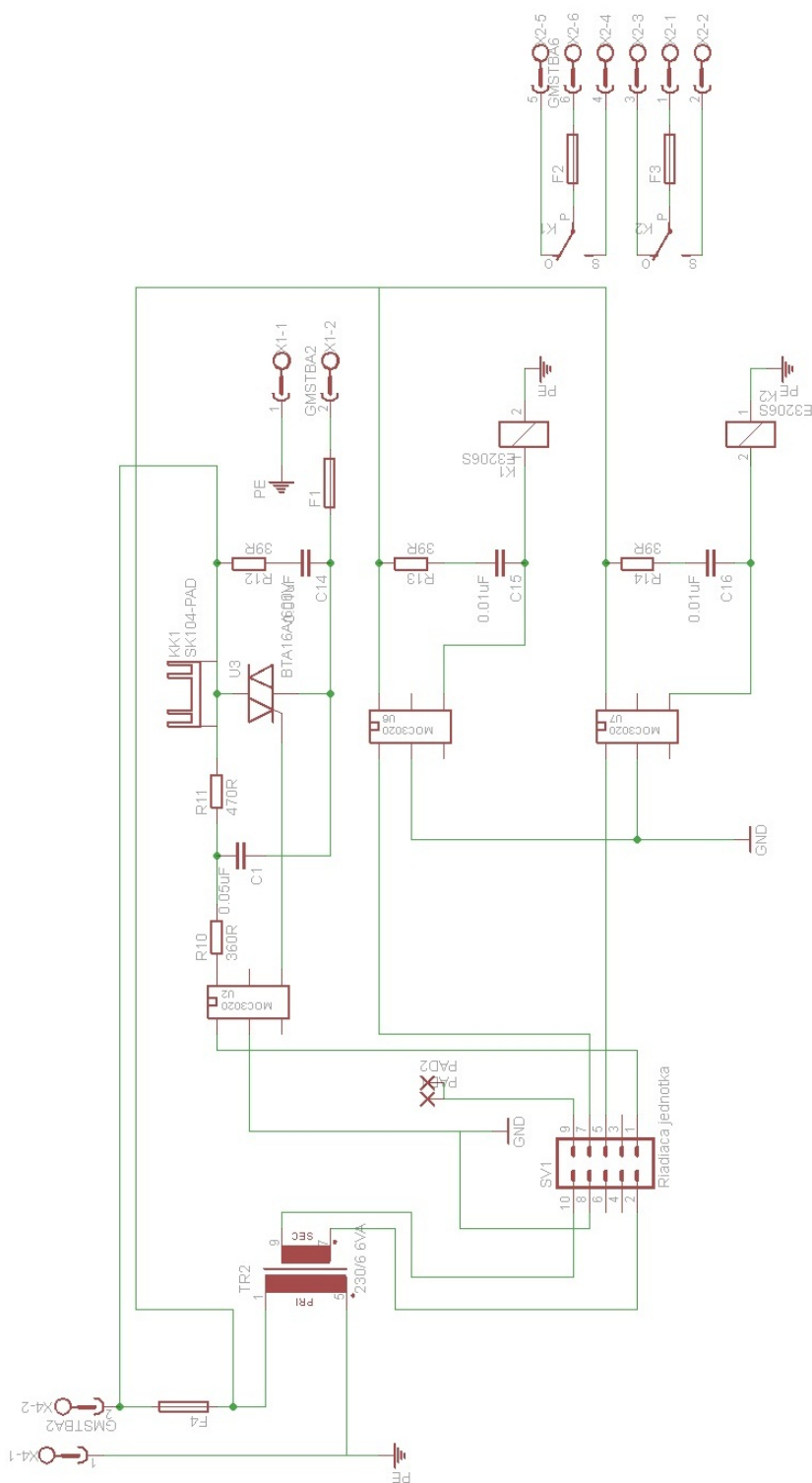
- Príloha č.1 - schéma výkonovej časti dosky
- Príloha č.2 - doska plošných spojov výkonovej časti
- Príloha č.3 - schéma riadiacej časti dosky
- Príloha č.4 - doska plošných spojov riadiacej časti
- Príloha č.5 - výpis zdrojového kódu teplotného čidla DS18B20
- Príloha č.6 - výpis zdrojového kódu pre obsluhu USART
- Príloha č.7 - výpis zdrojového kódu pre obsluhu LCD
- Priložené CD

Obsah CD:

- Aplikácie - obsahuje zdrojový kód celého projektu vrátane vytvorených knižníc
- Datasheety - obsahuje datasheety použitých súčiastok
- Výroba dosky - obsahuje materiály potrebné k výrobe dosky plošného spoja
- Text diplomovej práce - obsahuje finálny text diplomovej práce

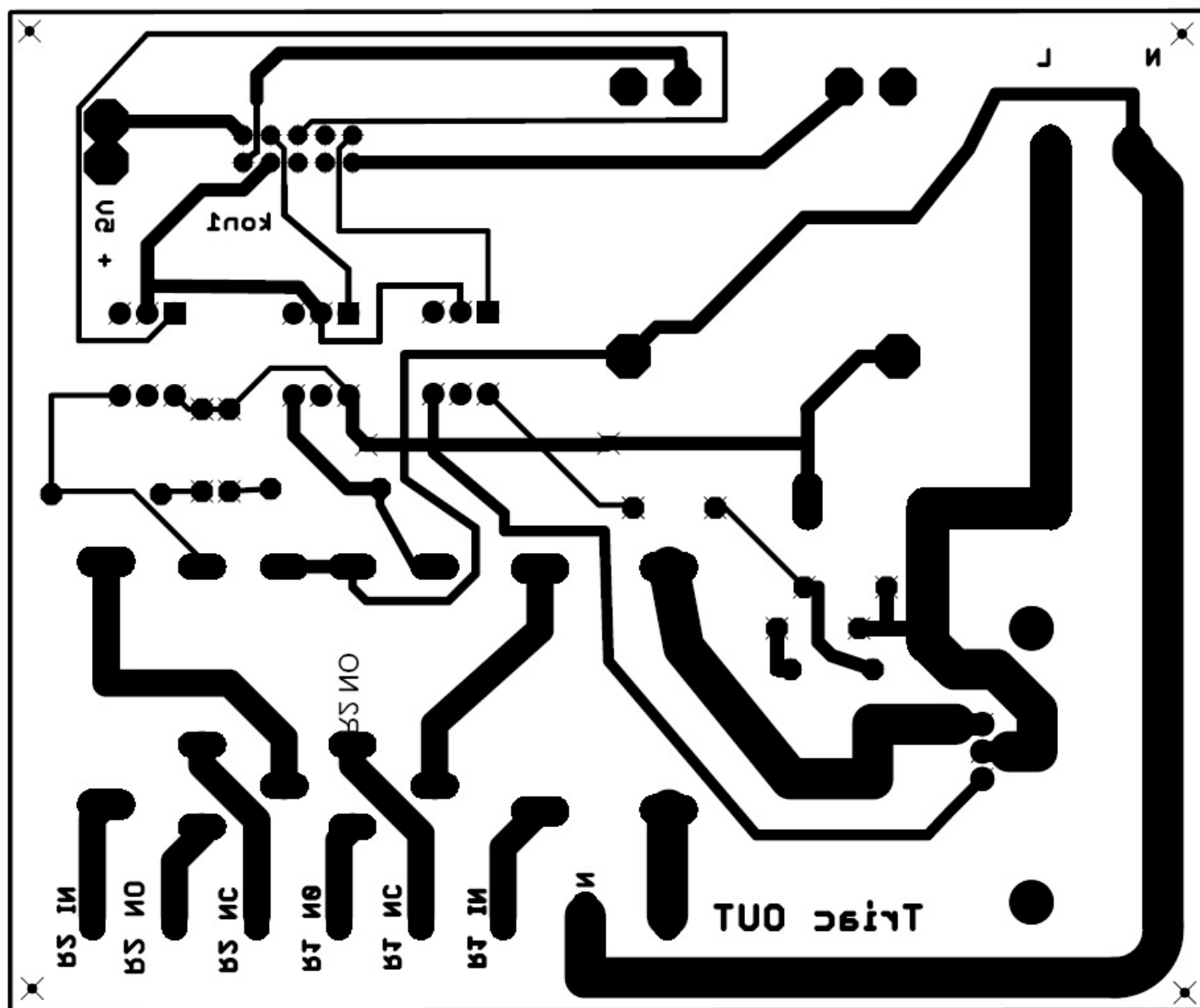
# 10. Prílohy

## 10.1. Príloha č.1 - schéma výkonovej časti dosky



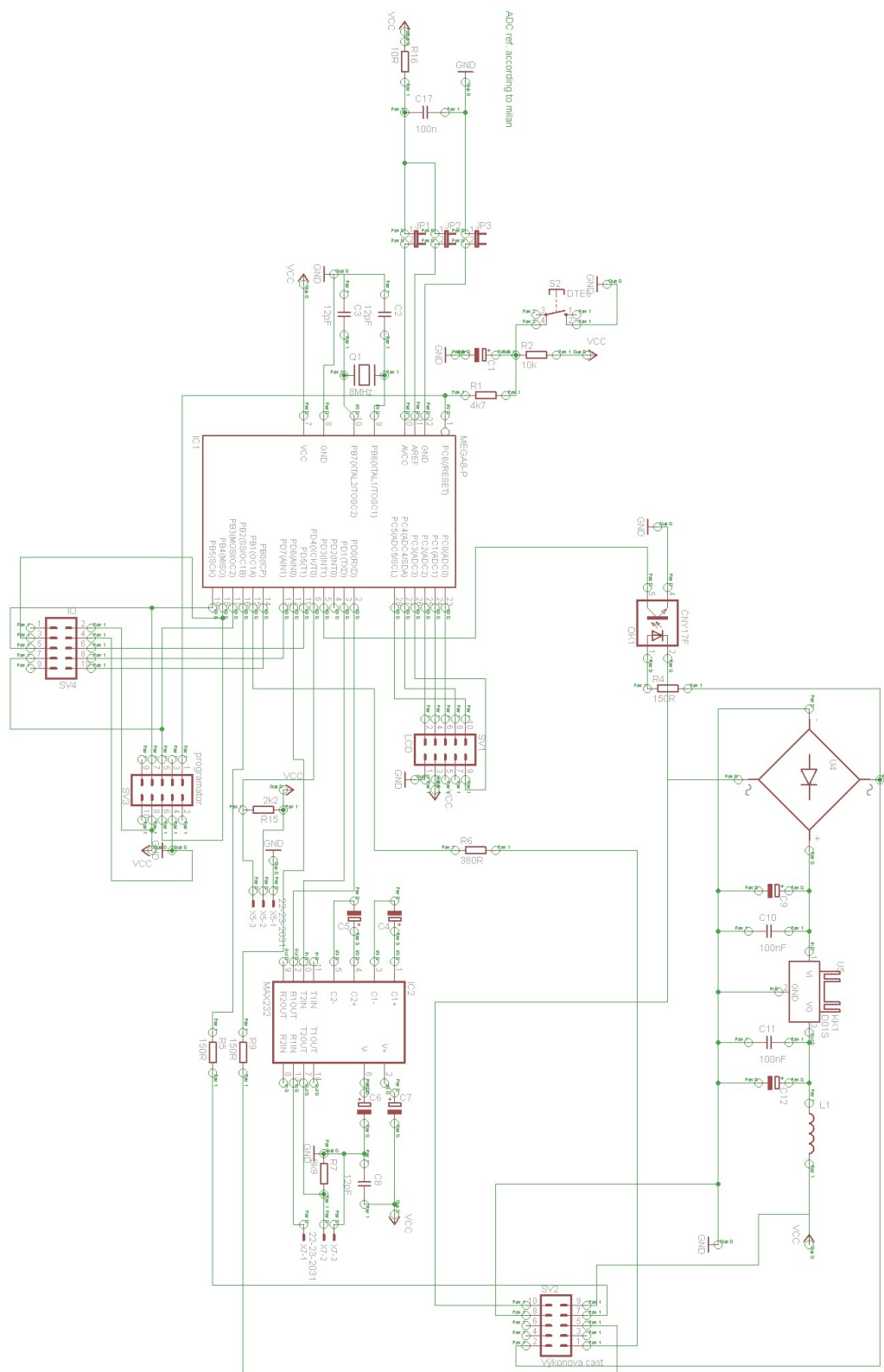
Obr. 10.1: schéma výkonovej časti dosky

## 10.2. Príloha č.2-doska plošných spojov výkonovej časti



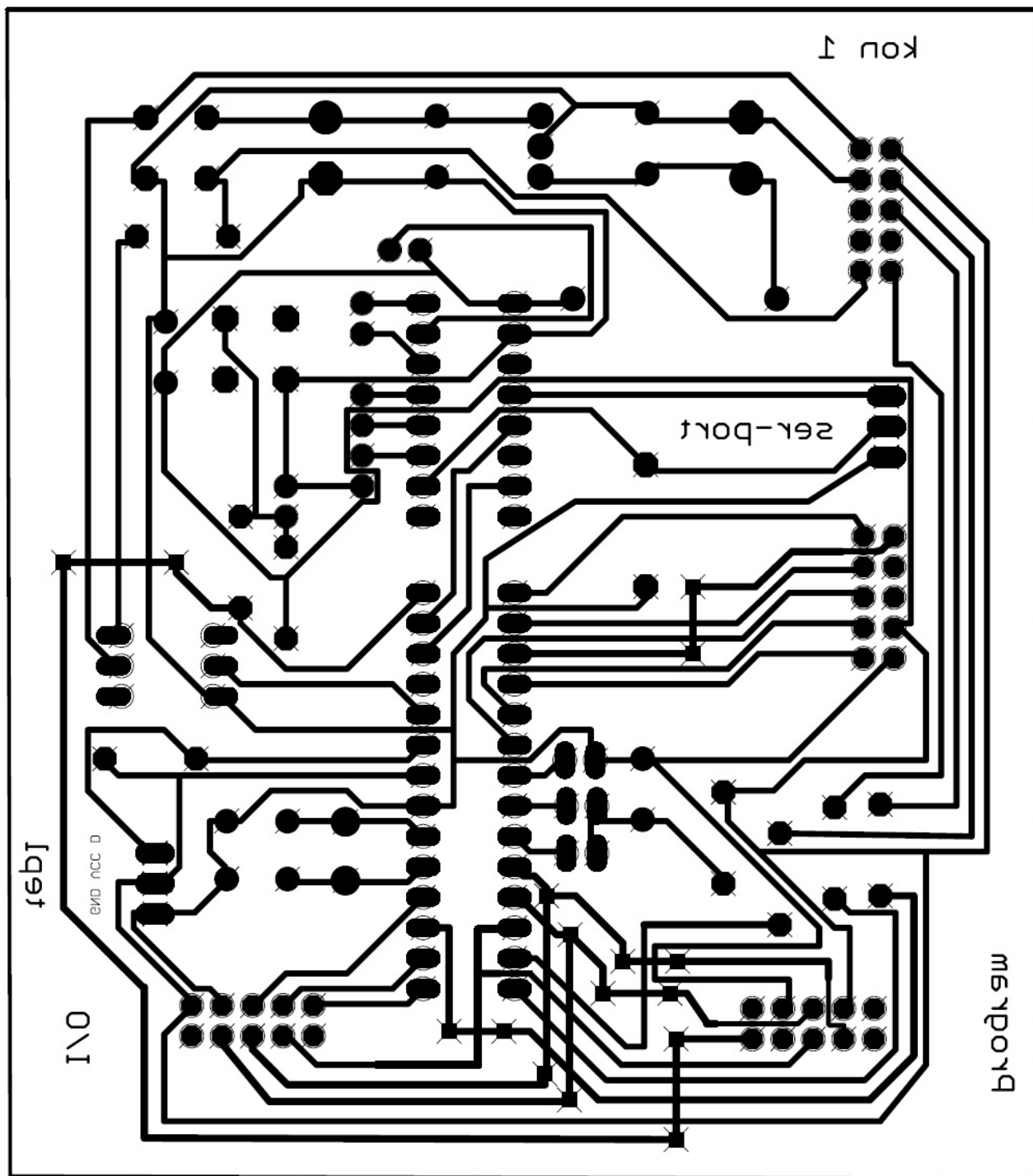
Obr. 10.2: doska plošných spojov výkonovej časti

### 10.3. Príloha č.3 - schéma riadiacej časti desky



Obr. 10.3: schéma riadiacej časti desky

## 10.4. Príloha č.4 - doska plošných spojov riadiacej časti



Obr. 10.4: doska plošných spojov riadiacej časti



## 10.5. Príloha č.5 - výpis zdrojového kódu teplotného čidla DS18B20

```

#ifndef __teplotne_cidlo_h
#define __teplotne_cidlo_h
#include <delay.h>
#include <mega8.h>
#include <stdlib.h>
#include <math.h>
#include <LCD_knih.h>

// provedie reset a test prezencneho signalu
#if funcused reset_prezenc
unsigned char reset_prezenc(void) {
    unsigned char pritomnost=1;                // vychodi navratova hodnota
    DDRD &= ~0x10;                            // vychodi stav sbernice
    delay_us(100);                            // pro ustaleni
    DDRD |= 0x10;                             // bus low
    delay_us(480);
    DDRD &= ~0x10;                            // uvolneni sbernice
    delay_us(70);                             // cekani na potvrzeni teplomerem
    if(PIND&0x10)
    {
        pritomnost=0;                        // pokud detekovana log.1, tak teplomer na sbernici n
        LCD4_Clear();
        LCD4_Home();
        LCD4_PutS("Cidlo neprit");
        return pritomnost;
    }
    delay_us(410);

    // vrati stav 1=teplomer pritomny, 0=teplomer nepritomny
}

```

## 10.5. PRÍLOHA Č.5 - VÝPIS ZDROJOVÉHO KÓDU TEPLOTNÉHO ČIDLA DS18B20

```
#endif

// posle na sbernici log.1
#if funcused zapis_log1
void zapis_log1(void) {
    DDRD |= 0x10;
    delay_us(6);
    DDRD &= ~0x10;
    delay_us(64);
}
#endif

// posle na sbernici log.0
#if funcused zapis_log0
void zapis_log0(void) {
    DDRD |= 0x10;
    delay_us(60);
    DDRD &= ~0x10;
    delay_us(10);
}
#endif

// precte jeden bit ze sbernice
#if funcused citaj_bit
unsigned char citaj_bit(void) {
    unsigned char hodnota=0;           // navratova hodnota
    DDRD |= 0x10;
    delay_us(2);
    DDRD &= ~0x10;
    delay_us(8);
    if(PIND&0x10) hodnota=1;           // test stavu zbernice a zapis log1
    delay_us(120);
    return hodnota; //vrati hodnota=0 alebo hodnota=1
}
```

```

}
#endif

// odeslne na sbernici jeden byte. Odesila se prvni LSB
#if funcused zapis_byte
void zapis_byte(unsigned char hodnota_bytu) {
    int n=8;
    while(n--) {
        if(hodnota_bytu & 1) zapis_log1(); else zapis_log0();
        hodnota_bytu >>= 1;
    }
}
#endif

// prijde ze sbernice jeden byte. Prijima jako prvni LSB.

#if funcused citaj_byte
unsigned char citaj_byte(void) {
    int n=8, hodnotaR=0;
    while(n--) {
        hodnotaR >>= 1; // bitovy posuv doprava
        if(citaj_bit()) hodnotaR |= 0x80; // nastaveni nejvyssiho bitu na 1
    }
    return hodnotaR;
}
#endif
#endif

```

## 10.6. Príloha č.6 - výpis zdrojového kódu pre obsluhu USART

```
\*Prevzate a upravene z http://code.google.com/p/reflowoven/source/browse/trunk/USART2.c?sp

#include <mega8.h>
// #include <macros.h>
#include <uart_def_I.h>
/* uart globals */

static unsigned char USART_RxBuf[USART_RX_BUFFER_SIZE];
static volatile unsigned char USART_RxHead;
static volatile unsigned char USART_RxTail;
static unsigned char USART_TxBuf[USART_TX_BUFFER_SIZE];
static volatile unsigned char USART_TxHead;
static volatile unsigned char USART_TxTail;
#if funcused Flush_UART_Buffer
void Flush_UART_Buffer(void){
    unsigned char x;
    unsigned char dummy;
    while ( UCSRA & (1<<RXC) ) dummy = UDR;
    /* Flush receive buffer */
    x = 0;
    USART_RxTail = x;
    USART_RxHead = x;
    USART_TxTail = x;
    USART_TxHead = x;
    USART_RxBuf[0]=0;
    USART_RxBuf[1]=0;
    USART_RxBuf[2]=0;
    USART_RxBuf[3]=0;
    USART_RxBuf[4]=0;
    USART_RxBuf[5]=0;
    USART_RxBuf[6]=0;
}
```



## 10.6. PRÍLOHA Č.6 - VÝPIS ZDROJOVÉHO KÓDU PRE OBSLUHU USART

```
UCSRB &= ~(1<<UDRIE);          /* Disable UDRE interrupt */
}

}

/* Read and write functions */
#if funcused UART_Receive
unsigned char UART_Receive( void )
{
    unsigned char tmptail;
    while (USART_RxHead == USART_RxTail); /* Wait for incomming data */
    tmptail = ( USART_RxTail + 1 ) & USART_RX_BUFFER_MASK; /* Calculate buffer index */
    USART_RxTail = tmptail;          /* Store new index */
    return USART_RxBuf[tmptail];     /* Return data */
}
#endif

\\prijem

#if funcused UART_VypisBufer
unsigned char UART_VypisBufer( unsigned char index)
{
    return USART_RxBuf[index];       /* Return data */
}
#endif

/* odosielanie */
#if funcused UART_Transm
void UART_Transm(unsigned char data )
{
    unsigned char tmphead;
    /* Calculate buffer index */
    tmphead = ( USART_TxHead + 1 ) & USART_TX_BUFFER_MASK; /* Wait for free space in buffer */
}
```

```

while ( tmphead == USART_TxTail );
USART_TxBuf[tmphead] = data;          /* Store data in buffer */
USART_TxHead = tmphead;              /* Store new index */
UCSRB |= (1<<UDRIE);                /* Enable UDRE interrupt */
}
#endif
//-----
#if funcused DataInReceiveBuffer0
unsigned char DataInReceiveBuffer0( void )
{
return ( USART_RxHead != USART_RxTail ); /* Return 0 (FALSE) if the receive buffer is
}
#endif
//-----
////////////////////////////////////
//vracia pocet aktualne obsadenych bajtov v bufere
#if funcused DataInReceiveBuffer0b
unsigned char DataInReceiveBuffer0b( void )
{
int tmp_ur;
if (USART_RxHead >= USART_RxTail)
tmp_ur = (USART_RxHead - USART_RxTail); /* Return 0 (FALSE) if the receive buffer is
else
tmp_ur = (USART_RxHead + (USART_RX_BUFFER_SIZE - USART_RxTail)); /* Return 0 (FALSE)
return (tmp_ur);
}
#endif

```

## 10.7. Príloha č.7 - výpis zdrojového kódu pre obsluhu LCD

\\*Prevzate a upravene z <http://read.pudn.com/downloads42/sourcecode/embed/144385/LCD4BIT.C>.

```
#include <mega8.h>
#include <delay.h>
#include <stdlib.h>
#include <math.h>
#define WASTE                waste_lcd()
/* pripojenie LCD k portu C */
#define LCD_E_SET            PORTC |= 0x20
#define LCD_E_CLR            PORTC &= ~0x20
#define LCD_RS_SET           PORTC |= 0x10
#define LCD_RS_CLR           PORTC &= ~0x10
#define DELAY                for (del = 0; del < 16; del++)
    long del;

#if funcused waste_lcd
void waste_lcd(void)
{
}
#endif

// zapisovanie dat na LCD
#if funcused LCD4_WR4bits
void LCD4_WR4bits(unsigned char c)
{
    c &= 0x0F;
    LCD_E_SET;
    WASTE;
    PORTC &= 0xF0;
    PORTC |= c;
    WASTE;
```



```

LCD_E_CLR;
WASTE;
}
#endif

// inicializacia LCD
#if funcused LCD4_Init
void LCD4_Init(void){
    delay_ms(15);                // Wait on power up 15mS
    LCD_RS_CLR;
    DELAY;
    LCD4_WR4bits( 0x03);         // Function Set 8bit
    delay_ms(5);
    LCD4_WR4bits( 0x03);         // Function Set 8bit
    delay_ms(5);
    LCD4_WR4bits( 0x03);         // Function Set 8bit
    delay_ms(5);
    LCD4_WR4bits( 0x02);         // Function Set 4bit
    delay_ms(5);
    LCD4_WR4bits( 0x02);
    LCD4_WR4bits( 0x08);
    DELAY;
    LCD4_WR4bits( 0x00);         // Display ON/OFF
    LCD4_WR4bits( 0x0C);         // Display ON, Curson ON, Blink OFF
    DELAY;
    LCD4_WR4bits( 0x00);         // Entry mode
    LCD4_WR4bits( 0x06);
    DELAY;
    LCD_RS_SET;
    delay_ms(5);
}
#endif

```

## 10.7. PRÍLOHA Č.7 - VÝPIS ZDROJOVÉHO KÓDU PRE OBSLUHU LCD

```
// smaze cele LCD
#if funcused LCD4_Clear
void LCD4_Clear(void){
LCD_RS_CLR;
DELAY;
LCD4_WR4bits( 0x00);    // Clear Display
LCD4_WR4bits( 0x01);
DELAY;
LCD_RS_SET;
delay_ms(5);
}
#endif

//posunie kurzor na zaciatok prveho riadku
#if funcused LCD4_Home
void LCD4_Home(void){
LCD_RS_CLR;
DELAY;
LCD4_WR4bits( 0x00);
LCD4_WR4bits( 0x02);
DELAY;
LCD_RS_SET;
delay_ms(5);
}
#endif

//posunie kurzor na zaciatok druheho riadku
#if funcused LCD4_Home2
void LCD4_Home2(void){
LCD_RS_CLR;
DELAY;
```

```

LCD4_WR4bits( 0x0C);    // Clear Display
LCD4_WR4bits( 0x00);
DELAY;
LCD_RS_SET;
delay_ms(5);
}
#endif

/*zapis dat */
#if funcused LCD4_PutC
void LCD4_PutC(unsigned char data)
{
LCD4_WR4bits( data >> 4);
DELAY;
LCD4_WR4bits( data);
DELAY;
}
#endif

#if funcused LCD4_PutC_Const
void LCD4_PutC_Const(const unsigned char data){
LCD4_WR4bits( data >> 4);
DELAY;
LCD4_WR4bits( data);
DELAY;
}
#endif

//zapise retazec znakov na LCD
#if funcused LCD4_PutS
void LCD4_PutS(unsigned char *data){
unsigned char c = 0;

```

## 10.7. PRÍLOHA Č.7 - VÝPIS ZDROJOVÉHO KÓDU PRE OBSLUHU LCD

```
while (data[c] != 0)
{
    LCD4_WR4bits( data[c] >> 4);
    LCD4_WR4bits( data[c++]);
    DELAY;
}
}
#endif

// Posun kurzora doprava o jeden znak
/*
#if funcused LCD4_GotoR
void LCD4_GotoR(int i)
{
    while (i>0)
    {
        LCD_RS_CLR;
        DELAY;
        LCD4_WR4bits(0x01);
        LCD4_WR4bits(0x04);
        DELAY;
        LCD_RS_SET;
        delay_ms(5);
        i=i-1;
    }
}
#endif

// Posun kurzora dolava o jeden znak

#if funcused LCD4_GotoL
```

```
void LCD4_GotoL(int i)
{
while (i>0)
{
    LCD_RS_CLR;
    DELAY;
    LCD4_WR4bits(0x01);
    LCD4_WR4bits(0x00);
    DELAY;
    LCD_RS_SET;
    delay_ms(5);
    i=i-1;
}
}
#endif
*/
```